

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Prototype de système expert pour la mise au point d'un patient comateux

Klein, Thierry

Award date:
1987

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

PROTOTYPE DE SYSTEME EXPERT

POUR LA MISE AU POINT

D'UN PATIENT COMATEUX

Année académique 1986-1987

Thierry Klein

Promoteur:
Professeur J.Fichefet

Mémoire présenté en vue
de l'obtention du grade
de Licencié et Maître
en Informatique

REMERCIEMENTS

Je voudrais remercier le Professeur J.Fichefet qui a dirigé ce mémoire avec gentillesse et attention ainsi que J.P.Leclercq qui m'a aidé à le terminer.

Je tiens à exprimer ma reconnaissance au Dr Saliez de la Clinique St Jean qui a suivi de près ce travail et a mis à ma disposition tous les moyens nécessaires à sa réalisation.

Je remercie enfin le Dr Van Keerbergen et toute l'équipe du service des Soins Intensifs pour leur accueil et leurs conseils relatifs à la partie médicale du mémoire.

TABLE DES MATIERES

INTRODUCTION

PARTIE I: CONCEPTS GENERAUX

1) INFORMATIQUE MEDICALE

1.1) Objectifs

1.2) Domaines d'application

2) RAISONNEMENT MEDICAL ET INFORMATIQUE

2.1) Raisonnement médical

2.1.1) La démarche du médecin

2.1.2) Connaissances et raisonnements médicaux

2.2) Modèles de diagnostic médical

2.2.1) Formalismes logiques

2.2.2) Méthodes statistiques

2.2.3) Méthode probabiliste (Théorème de Bayes)

2.2.4) Logique floue

2.2.5) Autres modèles

2.2.6) Systèmes experts (définition et principe)

2.3) Synthèse

3) SYSTEMES EXPERTS EN MEDECINE

3.1) Structure et fonctionnement

3.2) Représentation des connaissances et mécanismes d'inférence

3.2.1) Définition

3.2.2) Typologie des connaissances

3.2.3) Techniques de représentation

3.2.4) Synthèse

3.3) Exemples de systèmes experts médicaux

3.3.1) MYCIN

3.3.2) SPHINX

3.3.3) INTERNIST

3.3.4) CASNET

PARTIE II: APPLICATION

4) LE PROJET

4.1) Description générale

4.2) Objectifs

5) LE SUJET: MISE AU POINT D'UN PATIENT COMATEUX

5.1) Introduction

5.2) La démarche

5.3) Les premiers gestes

5.4) Causes principales de coma

5.5) Listes de signes par diagnostic

5.6) Critères de diagnostic

5.7) Glossaire

6) DEVELOPPEMENT

6.1) Introduction

6.2) Analyse d'opportunité

6.3) Spécification fonctionnelle

6.4) Conception

6.5) Codage

7) LE LANGAGE PROLOG

7.1) La programmation logique

7.2) Introduction au langage Prolog

7.3) Syntaxe du langage Prolog

7.4) Mécanismes de Prolog

7.4.1) Le chaînage arrière

7.4.2) L'unification

7.4.3) Le backtracking

7.4.4) Les primitives extra-logiques

7.5) Systèmes experts et Prolog

CONCLUSIONS

ANNEXE (Programme)

REFERENCES

INTRODUCTION

Depuis de nombreuses années, le médecin a utilisé des instruments et des machines pour l'aider dans sa pratique. Récemment, l'apparition de l'ordinateur a vu de nombreuses activités médicales se modifier et a permis le développement de nombreuses applications nouvelles. Parmi celles-ci, l'aide à la décision et au diagnostic est certainement à la fois l'une des plus difficiles à réaliser et celle qui produit le plus de réticences auprès des médecins. La raison en est probablement que l'on touche au côté "artistique" de la médecine (on dit souvent que la médecine est à la fois un art et une science), c'est à dire à l'intuition et à l'expérience comme facteurs de prise de décision. Il est donc difficile de les formaliser et on peut comprendre que prétendre réaliser une simulation informatique du raisonnement médical appelle de la méfiance de la part du corps médical.

De nombreuses réalisations, aides au diagnostic ou aides à la décision, existent depuis longtemps même si elles restent le plus souvent expérimentales et donc peu utilisées en pratique courante. La plupart d'entre elles sont basées sur les statistiques ou sur le calcul des probabilités. Leurs performances sont parfois remarquables mais leur limite principale est de ne pas tenir compte de la manière dont le médecin procède pour arriver à une conclusion (diagnostic ou prise de décision). En effet, un tel système produit un résultat au terme d'un calcul et en présence d'information complète; le médecin a donc tendance à le considérer comme une boîte noire dont le mode de fonctionnement est totalement étranger à sa propre démarche. Pour pallier à cet inconvénient, il apparaît donc qu'une autre approche s'avère nécessaire.

Récemment, un nouveau domaine de l'informatique est apparu et est connu sous le nom d'"intelligence artificielle". Sa principale caractéristique est de fournir à l'ordinateur des connaissances et, à un certain degré, des mécanismes de manipulation de celles-ci dans le but de le rendre capable de trouver une stratégie pour fournir un résultat. Il ne s'agit donc plus de coder un algorithme solution d'un problème mais de donner au système le moyen de "raisonner" à partir de ce qu'il sait. Cette discipline n'étant qu'à l'aube de son histoire, il faut éviter des considérations triomphalistes mais force est de constater qu'il s'agit d'une optique intéressante dans la modélisation du raisonnement.

Un des domaines d'application de l'"intelligence artificielle" est ce

que l'on appelle les systèmes experts, qui sont des programmes chargés de résoudre des problèmes habituellement confiés à des experts humains. Dans le domaine médical, en particulier dans l'aide au diagnostic, il en existe actuellement de nombreuses réalisations qui restent le plus souvent expérimentales malgré leurs bonnes performances et leur capacité à modéliser le raisonnement du médecin. Les caractéristiques de tels systèmes sont de manipuler des connaissances représentées de manière explicite, de les rendre extensibles et modifiables à souhait, d'expliquer le raisonnement suivi et de communiquer dans un langage proche du langage naturel. Par rapport aux techniques antérieures, le système expert est donc plus proche de l'utilisateur, plus facile à comprendre et plus souple. Il faut cependant insister sur le fait que l'on est loin d'avoir conçu un système acceptable en médecine courante. Cela tient probablement en grande partie à la diversité et à la complexité du raisonnement humain en général, du raisonnement médical en particulier.

C'est ce dernier aspect qui a été déterminant dans le choix du sujet de ce mémoire parce qu'au delà de la découverte d'une nouvelle technique informatique, la formalisation des connaissances et raisonnements médicaux impose une réflexion en profondeur sur leur nature. D'autre part, les systèmes experts médicaux étant encore à ce jour de nature expérimentale, il semblait intéressant d'étudier de manière concrète les difficultés de leur réalisation tout en essayant de faire entrevoir leurs possibilités par la mise au point d'un prototype. L'équipe informatique de la clinique St Jean à Bruxelles a la première proposé un sujet de mémoire dans le domaine de l'aide à la décision (sans encore parler de système expert) dans le double but de découvrir de nouvelles méthodes informatiques et de stimuler l'intérêt du corps médical. Le service des soins intensifs de cette clinique a été choisi comme lieu de développement et c'est en collaboration avec les médecins de ce service que le sujet a été défini de manière plus précise. Le choix s'est porté sur la réalisation d'un système expert pour aider à la décision et au diagnostic dans la mise au point d'un patient comateux en urgence. Dans cette situation, le conseil d'un expert peut se révéler utile et le problème, tout en étant pas trop large, est loin d'être trivial. En effet, il faut envisager de multiples alternatives englobant de

nombreuses connaissances et nécessitant des stratégies de recherche. Toutes ces caractéristiques sont propices au développement d'un système expert.

PLAN DU MEMOIRE

Le travail comporte deux grandes parties.

-Dans la première, les concepts généraux relatifs à la simulation informatique du raisonnement médical seront développés. Cette section comprend les trois premiers chapitres du mémoire.

Le premier servira d'introduction et traitera des objectifs et des domaines d'application de l'informatique médicale.

Dans le deuxième chapitre, nous aborderons le coeur du sujet, à savoir l'étude en profondeur de la démarche intellectuelle du médecin. Nous étudierons ensuite les différents modèles qui permettent de formaliser le diagnostic médical. C'est ainsi que nous passerons en revue les méthodes basées sur la logique, les statistiques et les probabilités. En mettant en évidence leur intérêt et limites respectifs, nous les utiliserons comme point de comparaison pour l'étude des systèmes experts.

Ces derniers feront l'objet du troisième chapitre. Nous commencerons par en décrire la structure et le fonctionnement avant d'aborder les deux grandes difficultés dans la mise au point de tout système expert, à savoir la représentation des connaissances et le choix d'un mécanisme d'inférence. Nous clôturerons cette première partie par l'exposé de quatre exemples de systèmes experts médicaux.

-La deuxième partie sera consacrée à la réalisation du prototype de système expert.

Nous commencerons, au chapitre 4, par expliquer comment est née l'idée de ce projet et par décrire son environnement avant d'en préciser les grands objectifs.

Le chapitre 5 décrit en détail le sujet du point de vue médical. L'accent est mis sur la démarche du médecin en urgence (étude des premiers gestes) et sur les critères de décision et de diagnostic.

Les différentes étapes de développement du systèmes feront l'objet du sixième chapitre. Nous essayerons de refléter le raisonnement suivi en

justifiant le plus souvent possible les décisions prises et en explicitant les hypothèses le cas échéant.

Enfin, nous expliquerons dans les grandes lignes au chapitre 7 les caractéristiques du langage Prolog, utilisé pour réaliser le système expert. Ceci permettra au lecteur non initié à ce langage de comprendre le code du programme donné en annexe. D'autre part, une partie spéciale essayera de montrer l'intérêt et les limites de l'utilisation de Prolog dans le développement d'un système expert.

PARTIE I : CONCEPTS GENERAUX

1) INFORMATIQUE MEDICALE

1.1) Objectifs

L'informatique médicale a pour objet d'utiliser les ordinateurs pour gérer rationnellement les informations relatives à la santé des patients.

Cette discipline s'est considérablement développée ces dernières années particulièrement au sein des hopitaux.

L'outil informatique y a été vu au départ essentiellement comme une aide aux tâches administratives, dans un but de rationalisation et d'économie. Le corps médical a longtemps considéré qu'il n'était pas concerné par cette évolution technique jusqu'au moment où l'on s'est aperçu de l'utilité de l'ordinateur dans la pratique du médecin.

Bien que l'informatique médicale n'est qu'à l'aube de son histoire, elle permet déjà d'offrir une aide précieuse dans un champ d'application de plus en plus large. On peut en escompter une modification de la pratique médicale dans le sens d'une amélioration de la qualité des soins. En effet, la gestion informatisée des informations relatives au patient permettent un meilleur suivi et les multiples applications médicales, à de nombreux échelons de la prise en charge du malade, offrent une efficacité accrue ou de nouvelles possibilités.

Il est probable que, dans l'avenir, les médecins seront amenés à utiliser de plus en plus l'ordinateur comme instrument de travail quotidien. L'informatique médicale a donc aussi pour objectif d'être au service du médecin dans son activité (dossier médical, diagnostic, communication de résultats,...).

Afin de mieux percevoir l'impact de l'informatique sur la médecine, il est bon de faire un rapide inventaire des principaux domaines d'application, ce qui fait l'objet de la partie qui suit.

1.2) DOMAINES D'APPLICATION

Les domaines de l'informatique médicale correspondent aux quatre grands objectifs de la médecine [ROGER 79]:

1° LES SOINS:

- diagnostic: .aide à la décision
 - .traitement des résultats d'examens
 - .imagerie médicale
- traitement: .médicaments (interaction, p.ex.)
 - .calcul de doses en radiothérapie
 - .soins intensifs (monitoring)
- évaluation: .résultats atteints
 - .procédures

2° LA RECHERCHE:

- documentation automatique: .bibliographie
 - .nomenclatures
 - .dossier médical
- biostatistiques: .enquêtes épidémiologiques
- simulation: .modèles stochastiques

3° L'ENSEIGNEMENT:

- enseignement assisté par ordinateur (en interactif)

4° L'INSERTION SOCIALE

- gestion hospitalière: .comptabilité
 - .facturation
 - .personnel
 - .dossier médical
 - .gestion des résultats
 - .prescription médicale
- santé publique .planification et choix budgétaires

Parmi ces domaines d'application, l'aide à la décision a fait couler beaucoup d'encre et a passionné nombre de chercheurs ces dernières années. L'idée de modéliser le diagnostic médical sur ordinateur peut être expliquée par plusieurs raisons.

D'une part, une approche systématique se basant sur de nombreuses connaissances permet d'espérer une amélioration de la qualité du diagnostic.

D'autre part, sa reproductibilité pourrait être meilleure si l'on parvient, par l'informatique, à fournir également les critères expliquant les résultats.

On augmenterait donc la compréhension du raisonnement médical et du processus de prise de décision, ce qui servirait la cause de l'enseignement médical.

En fin de compte, c'est le patient lui-même qui devrait profiter d'une meilleure qualité des soins.

L'idée d'un ordinateur remplaçant le médecin dans la démarche diagnostique et dans la prise de décisions s'est pourtant vite révélée un mythe. La raison principale réside dans la nature complexe du raisonnement médical qui sera étudié dans le chapitre suivant.

2)RAISONNEMENT MEDICAL ET INFORMATIQUE

2.1) LE RAISONNEMENT MEDICAL

2.1.1)La démarche du médecin

Il est intéressant d'étudier la manière dont un médecin arrive à poser un diagnostic et à prendre des décisions. Face à un malade, quel est le comportement intellectuel du médecin?

Un processus en plusieurs étapes

Il est difficile de donner une réponse stéréotypée à cette question, l'attitude pouvant varier d'un malade à l'autre et d'un médecin à l'autre. On peut cependant schématiquement résumer la démarche en plusieurs étapes [FIESCHI 86a]:

1)Receuil des symptômes et des signes cliniques

Le patient est interrogé et examiné.

2)Evocation d'hypothèses

Il a été montré que le médecin en génère 4 ou 5 au maximum, l'intuition et la détermination de faits saillants jouant un rôle important. Il les classe selon leur importance relative et oriente sa recherche en conséquence. Toutes les informations ne sont donc pas considérées en vrac mais sont organisées de manière à focaliser la recherche selon les hypothèses.

3) Evaluation des hypothèses

Pour confirmer ou infirmer une hypothèse, des informations supplémentaires sont souvent nécessaires, par le biais par exemple d'examens complémentaires (analyse de laboratoire, radiographie,...). La recherche est maintenant sélective. Petit à petit, une image cohérente de l'état du patient se construit.

D'autres facteurs peuvent intervenir dans cette évaluation. Citons l'intime conviction du médecin qui peut difficilement être expliquée formellement mais qui se base sur l'expérience professionnelle. Un modèle physiopathologique, décrivant les mécanismes d'apparition de la maladie envisagée, permet également d'interpréter les signes trouvés et de chercher les éventuels chaînons manquants.

4) Traitement et pronostic

Le(s) diagnostic(s) retenu(s), un traitement est instauré et le pronostic est déterminé le cas échéant.

Les quatre étapes décrites ci-dessus doivent être considérées comme des points de passage obligés et non comme une séquence figée. En effet, les premières hypothèses envisagées orientent l'acquisition des informations; de même, un traitement peut être instauré sans diagnostic préalable.

Stratégies de raisonnement

La démarche intellectuelle du médecin, pour passer par ces différentes étapes utilise, souvent inconsciemment, des stratégies de raisonnement qui peuvent être essentiellement de trois types [WILLIAMS 82]:

La première est issue de la théorie du "GESTALT" et est une perception intuitive globale du problème. Le diagnostic est parfois posé sur base d'une impression du clinicien, "flairant" une maladie. Ceci ne veut absolument pas dire que l'hypothèse est évoquée par hasard ni qu'elle est

imprécise mais la manière dont elle a été générée fait appel à des mécanismes inconscients élaborés au fil de l'expérience du praticien.

La seconde approche est de nature algorithmique, suivant un organigramme de décisions séquentielles. Il faut dans ce cas que les connaissances soient non ambiguës et qu'elles puissent se rattacher à un contexte particulier bien délimité. Ce processus de raisonnement nécessite des connaissances précises à des noeuds de décision sans quoi il y a blocage. D'autre part, il n'y a pas de possibilité de considérer les signes dans leur ensemble puisqu'ils sont évalués les uns après les autres.

Enfin, l'approche hypothético-déductive permet de formuler des hypothèses qui orientent la recherche. Cette méthode a l'avantage de pouvoir se concentrer sur un nombre restreint d'informations à la fois même si le problème du diagnostic différentiel se confine parfois à l'explosion combinatoire. Cette dernière approche est plus générale: le médecin peut utiliser la perception intuitive et la démarche algorithmique au sein d'un raisonnement hypothético-déductif.

Il y a donc souvent combinaison des trois techniques pour arriver au résultat, pour résoudre l'"énigme". Il s'agit bien en effet d'un raisonnement investigatif: étant donné des signes, établir le diagnostic. Il est intéressant de constater que l'enseignement médical procède de la manière inverse: étant donné une maladie, décrire les signes.

Il ne faudrait cependant pas considérer ce processus comme un raisonnement parfaitement rigoureux basé sur des déductions à partir des informations disponibles pour arriver à la solution. La démarche du médecin est sensiblement différente.

En effet, on ne peut résumer la situation par la recherche d'un seul diagnostic, inévitable. De plus, plusieurs auteurs [KASSIRER 78] [PAUKER 76] ont montré que les hypothèses avancées précocément par le clinicien sont suggérées à partir de très peu de données considérées comme marquantes. Le flair clinique joue là un rôle important.

Il convient d'autant plus de nuancer que les raisonnements peuvent changer d'un praticien à l'autre. C'est ainsi qu'il apparaît que les jeunes médecins adoptent une attitude différente de celle des cliniciens expérimentés. Les premiers ont tendance à aborder le problème de manière

structurée, en essayant de se cantonner dans une approche algorithmique, tandis que les cliniciens expérimentés privilégie une recherche où plusieurs hypothèses sont évaluées en parallèle.

Une question complexe

La description de la démarche intellectuelle du médecin qui précède peut paraître complexe et difficilement systématisable. En fait, c'est bien le cas et l'on peut s'interroger sur les raisons de cet état de fait. Pourquoi n'est-il pas possible d'avoir des schémas de raisonnement standards en présence de certains signes, des algorithmes de décisions répondant à la majorité des situations rencontrées?

Pour répondre à cette question, il faut se rendre compte de la diversité et de la complexité du raisonnement médical. La nature des données médicales est très variée et les inférences autorisées le sont aussi. C'est pourquoi, cette question sera abordée de manière détaillée dans la partie qui suit. Nous essayerons de systématiser les différents types de connaissance et de raisonnement.

2.1.2) Connaissances et raisonnements médicaux

Les connaissances et raisonnements médicaux doivent être étudiés ensemble car il existe une interdépendance entre les deux. On ne peut séparer l'objet (connaissance) de l'action (raisonnement) sur lequel elle porte. Les inférences autorisées sont fonction de la nature des données même si une même connaissance peut être exploitée par plusieurs types de raisonnement.

Catégories de raisonnement

De façon systématique mais sans prétention d'exhaustivité, nous allons présenter quelques catégories de raisonnements utilisés par le médecin [CHOURAKI 85]:

1) Raisonnement déductif à partir de connaissances exclusivement vraies ou fausses

Il s'agit d'un raisonnement catégorique que l'on pourrait formaliser selon la logique.

Soient: S_1, S_2, \dots, S_n les signes observés par le médecin lors de l'examen du patient,

H_1, H_2, \dots, H_m les hypothèses diagnostiques qu'il formule sur base de ses observations,

Les schémas d'inférence sont [FAGOT LARGEAULT 85]:

.Le modus ponens

[(S-

$\rightarrow H)$ et $S] \rightarrow H$

par exemple: [(signe de Köplik \rightarrow rougeole) et signe de köplik] \rightarrow rougeole

Si la présence d'un signe de Köplik (tache rouge centrée par un point blanc, sur la face interne des joues) implique que le patient a une rougeole, et que le signe de Köplik est présent, alors le patient a une rougeole. Si le signe de Köplik est parfaitement spécifique (pathognomonique) de la

rougeole (c'est le cas), et que l'enfant a un signe de Köplik, alors il a une rougeole.

.Le modus tollens

$[(H \rightarrow S) \text{ et non } S] \rightarrow \text{non } H$

par exemple: [(eczéma \rightarrow démangeaison) et non démangeaison-
 \rightarrow non eczéma

Si tout eczéma démange et que cette lésion ne démange pas, alors ce n'est pas un eczéma. Si le prurit est un signe absolument sensible dans l'eczéma (c'est à dire qu'il se retrouve à coup sûr dans tout eczéma, ce qui le cas), et que le patient n'a pas de prurit, alors la lésion qu'il présente n'est pas une lésion eczémateuse.

Les signes parfaitement spécifiques (comme les tâches de Köplik dans la rougeole) sont rares mais précieux car ils permettent de poser un diagnostic de façon catégorique (par modus ponens). De même, les signes absolument sensibles permettent, quand ils sont absents, de rejeter avec certitude une hypothèse.

La stratégie du raisonnement, lorsqu'il est applicable, pourrait se résumer en deux étapes. La première est la réduction du nombre d'hypothèses possibles par la recherche de signes parfaitement sensibles et manquants. La seconde est la confirmation des hypothèses retenues par des signes spécifiques.

Malheureusement, il est exceptionnel de pouvoir disposer de symptômes totalement sensibles ou spécifiques. C'est pourquoi il faut tenir compte de faux positifs (présence d'un signe sans la maladie) qui affecte la spécificité et de faux négatifs (absence du signe en présence de la maladie) qui diminue la sensibilité.

2) Raisonnement à partir de données incertaines et/ou imprécises

Une connaissance est dite incertaine si on ne peut établir, à partir des informations disponibles, si elle est vraie ou fausse bien qu'elle aie avec certitude l'une de ces deux valeurs de vérité. Une donnée est imprécise dans la mesure où elle ne permet pas de déterminer de manière unique une valeur de vérité.

On rencontre ces deux situations en médecine courante où les patients présentent des plaintes souvent imprécises ou les données recueillies ne permettent pas de déterminer la présence ou non d'un signe important. Le symptôme de "chaleur excessive", par exemple, est une information à la fois incertaine et imprécise pour la recherche d'une fièvre éventuelle.

D'autre part, il faut bien reconnaître que les modèles physiopathologiques, expliquant les mécanismes d'apparition des maladies, sont encore trop souvent incertains et/ou imprécis.

3) Raisonnement à partir de données incomplètes

Il s'agit de la situation où il manque des données pour tirer une conclusion. On est donc amené à introduire des hypothèses sur les valeurs manquantes. C'est ce type de démarche que rencontre le médecin et qui a été longuement explicitée dans l'approche hypothético-déductive décrite en 2.1.1.

4) Raisonnement à partir de données partiellement inconsistantes

Un ensemble de connaissances précises et certaines peut être inconsistant si on peut trouver par des schémas d'inférence valides des propositions contradictoires. La médecine est dans une certaine mesure une base inconsistante quand il y a désaccord sur un modèle physiopathologique ou sur le bien fondé d'un traitement.

5) Raisonnement qualitatif

Ce type de raisonnement permet de savoir, de façon certaine, si des paramètres sont dans une plage déterminée sans s'attacher à la valeur précise de ceux-ci. Le médecin retient les dosages anormaux (augmentés ou diminués) relevés dans une analyse de sang plutôt que les résultats chiffrés. Tout au plus nuancera-t'il la modification (légèrement, fortement augmenté,...) pour disposer d'une information suffisante pour interpréter et prendre des décisions.

6) Raisonnement en présence de connaissances évolutives

Dans ce cas, il convient de remettre en cause des informations validées antérieurement. L'évolution d'une maladie, de résultats d'examens complémentaires, rentrent dans ce cadre.

7) Raisonnements plausibles

Les schémas d'inférence plausibles sont aussi appelés non rigoureux car on n'a pas la certitude que si les expressions de départ sont vraies, les expressions résultantes le sont encore.

Un exemple caractéristique est le raisonnement par analogie. Etant donné les propriétés P et Q qui sont satisfaites dans une situation S1 et une propriété P' qui est satisfaite dans une situation S2, on infère qu'une propriété Q', qui est dans le même rapport à Q que P' l'était à P, devrait être satisfaite dans une situation S2. Le clinicien ne fait rien d'autre en inférant une hypothèse diagnostique en comparant l'ensemble des signes présents avec les symptômes connus d'une maladie précise.

8) Métaraisonnement

On peut être amené à évaluer, piloter, expliquer son raisonnement, ce qui également utilisé par le clinicien qui essaie de mettre en ordre toutes les pièces du puzzle lors d'un diagnostic différentiel. Il s'agit donc d'un raisonnement sur le raisonnement.

3) SYSTEMES EXPERTS EN MEDECINE

Avant d'étudier les réalisations en médecine, nous analyserons la structure et le fonctionnement d'un système expert (SE) ainsi que les techniques de représentation existantes. Nous terminerons le chapitre par l'exposé d'exemples de SE médicaux.

3.1) STRUCTURE ET FONCTIONNEMENT

STRUCTURE

Un SE est constitué de cinq modules:

- une base de connaissances
- un moteur d'inférence
- un module d'acquisition de connaissances
- les modules d'entrée et de sortie
- une base des faits

La base de connaissances reprend les informations médicales provenant à la fois de la théorie médicale (connaissance scientifique) et de l'expérience de l'expert (connaissance empirique). Dans la plupart des SE, elles sont exprimées sous forme de deux types de représentation: les faits élémentaires (valeur d'une glycémie normale, par exemple) et les règles ("si la lésion dermatologique ne démange pas, alors il ne s'agit pas d'un eczéma", par exemple). Ces unités d'informations sont le plus souvent stockées de manière descriptive, sans penser à la manière dont elles seront utilisées (représentation dite déclarative). Cette structure modulaire facilite les modifications de la base de connaissances et assure l'indépendance des connaissances par rapport au mode de raisonnement.

Le moteur d'inférence permet au système expert de raisonner en manipulant les données du problème et la base de connaissance pour élaborer des conclusions.

Le système apprend grâce au module d'acquisition des connaissances qui, comprenant un langage proche du langage naturel, interprétera la phrase pour représenter sa sémantique dans un formalisme symbolique. C'est de cette façon que l'on aboutit à un ajout d'information à la base de connaissance.

Pour rendre cet ensemble opérationnel, il faut encore prévoir les modules d'entrée et de sortie qui opèrent le plus souvent par l'intermédiaire respectivement d'un clavier et d'un écran vidéo.

Les données spécifiques du problème posé sont rassemblées dans une base de faits. Dans le cas du diagnostic médical, il s'agira des informations issues de l'interrogatoire du patient, des symptômes, des signes cliniques et éventuellement des résultats des examens complémentaires (biologie, radiographie,...).

FONCTIONNEMENT

Le moteur d'inférence va essayer d'appliquer ses règles de connaissance aux faits entrés. Il peut demander par le module de sortie des informations complémentaires qui lui manquent pour évaluer des hypothèses. Ces questions mettent à contribution l'utilisateur qui peut ainsi suivre pas à pas le raisonnement. Le SE doit être en mesure de donner des explications sur le bien fondé à la fois de ses questions et de ses conclusions qu'il fournit par le module de sortie.

D'une manière caricaturale, on pourrait donc dire qu'un SE présente un comportement apparenté à l'intelligence dans la mesure où, à partir de ses connaissances et de ses capacités de raisonnement, il cherche à proposer des solutions via un dialogue avec l'utilisateur. On retrouve les principales caractéristiques de la démarche du médecin décrites dans le chapitre précédent.

La structure de base et le principe de fonctionnement tels qu'ils viennent d'être exposés pourraient donner l'impression que les SE ont permis de modéliser les connaissances et raisonnements médicaux là où les autres techniques avaient échoué. Il convient de nuancer tout de suite cette impression en mettant en évidence les techniques de représentation des connaissances et de formalisation du raisonnement utilisés par les SE. Cette étude, qui va permettre de montrer qu'on est fort loin d'avoir trouvé une solution unique et idéale à ces problèmes, fera l'objet de la partie qui suit (partie 3.2). Nous développerons donc deux aspects soulevés dans l'étude de la structure d'un SE (base de connaissance et moteur d'inférence).

3.2)REPRESENTATION DES CONNAISSANCES ET MECANISMES D'INFERENCE

3.2.1)DEFINITION

L'intérêt des SE a été surtout d'exploiter des connaissances explicites d'un domaine spécifique. Mais qu'est ce qu'une connaissance? Information et connaissance sont-elles des synonymes? Comme le souligne Gallaire [GALLAIRE 85], il convient de distinguer les deux termes. Un fichier, un livre ou un article contiennent des informations. On ne peut cependant affirmer qu'un livre "connait" ce qu'il renferme, car connaître suppose des mécanismes de mémorisation et de raisonnement sur des connaissances mémorisées. C'est ce qui différencie une base de données et une base de connaissances. La mise en exergue de cette différence est importante car elle constitue une illustration du lien qui existe entre une connaissance et la manière dont elle sera manipulée. Représentation des connaissances et mécanismes d'inférence sont souvent indissociables; c'est la raison pour laquelle nous les étudierons ensemble dans ce chapitre.

3.2.2) TYPOLOGIE DES CONNAISSANCES

Celles-ci sont nombreuses et diverses et bien qu'il n'existe pas de consensus relatif à leur classification, il est intéressant de montrer leur diversité avant d'étudier le problème de leur représentation. C'est ainsi que l'on peut définir [VAN LAMSWEERDE 86] différents points de vue pour aborder la définition des connaissances dans le cadre de la mise au point d'un SE:

1) Fonctionnalités des connaissances

- résolution de problème
- communication avec l'utilisateur
- acquisition de nouvelles connaissances
- explication du raisonnement
- vérification de la base
- conduite du moteur d'inférence

2) Types de connaissance

connaissances relatives à des:

- objets, concepts, relations, propriétés
- actions, événements
- axiomes, lois
- stratégies de résolution, heuristiques (techniques qui aident à la découverte d'une solution mais sans garantie de succès)
- métaconnaissances (connaissances relatives à des connaissances qui ont pour objet d'orienter la recherche d'une solution.)

3) Formes de connaissance

- certaine/incertaine
- précise/imprécise
- complète/incomplète
- plausible
- évolutive/non évolutive

4) Manipulations de connaissance

- acquisition
- restitution
- recherche
- types de raisonnement pour l'obtention de nouvelles connaissances:
 - . formel (chaînage, unification syntaxique)
 - . procédural (procédure d'acquisition de connaissances)
 - . par analogie
 - . par généralisation
 - . par défaut (une connaissance est supposée vraie tant que le contraire n'a pas été prouvé)
 - . métaraisonnement (raisonnement sur des métaconnaissances pour piloter le raisonnement effectif)

5) Modes de structuration des connaissances

- classification (relation "occurrence de")
- agrégation/décomposition (relation "est partie de")
- généralisation/spécialisation (relation "est un")

Cette typologie détaillée des connaissances appelle des commentaires.

Tout d'abord, la diversité et la complexité qu'elle illustre fait de la représentation des connaissances un problème majeur qui sera étudié dans la partie qui suit (3.2.3).

Ensuite, nous retrouvons dans les formes de connaissances des notions déjà développées dans l'étude du raisonnement médical (2.1). Ce fait confirme l'interdépendance entre une forme de connaissance et un type de raisonnement même si le principe même des SE est de représenter dans une approche déclarative les deux notions séparément (base de connaissance, moteur d'inférence). Autrement dit, le mode de représentation des connaissances définit souvent le type d'utilisation que l'on peut en faire.

Enfin, le propre d'un système expert étant de raisonner, le SE doit non seulement posséder des connaissances de définition mais aussi des connaissances de l'expert qui lui permettent de déduire de nouvelles informations déduites des données initiales du problème. C'est ce qui est exprimé dans les formes de manipulation des connaissances (raisonnements pour obtenir des nouvelles connaissances).

3.2.3) TECHNIQUES DE REPRESENTATION

La représentation des connaissances, formalisme servant de support aux phénomènes étudiés, a pour but de rendre les connaissances traitables par un ordinateur.

L'alternative souvent avancée est le choix entre représentation procédurale et déclarative. Dans la première, la connaissance indique explicitement comment elle est utilisée et est donc représentée par l'emploi de procédures. Dans la deuxième, il y a séparation entre la connaissance et son mode d'utilisation. La représentation déclarative a l'avantage de pouvoir modifier indépendamment les connaissances ou le raisonnement. C'est donc l'option choisie dans la plupart des SE et que l'on retrouve typiquement dans les règles de production.

Nous examinerons successivement la logique des propositions et la logique des prédicats du premier ordre, les systèmes de production et les objets structurés. Il ne s'agit pas ici de procéder à une revue exhaustive des modes de représentation mais bien de donner les principes

généraux de quelque uns d'entre eux qui sont utilisés dans les systèmes experts médicaux décrits dans la partie suivante (3.3:exemples de systèmes experts).

1.Logique des propositions et logique des prédicats

Ces types de représentation se basent sur la logique mathématique qui définit un système formel comme un ensemble de trois éléments:

- 1°un langage (avec une syntaxe)
- 2°une interprétation des formules (qui expriment la sémantique)
- 3°une théorie des preuves (modes de dérivation de formules valides syntaxiquement à partir d'autres).

1.1.Logique des propositions

Formalisme

Dans la logique des propositions, le langage consiste en des propositions (suite de symboles) simples ou composées (avec des connecteurs tels que \wedge (et logique), \vee (ou logique), \neg (négation logique), \rightarrow (implication), \leftrightarrow (équivalence)).

Une proposition simple est soit vraie soit fausse (par exemple, le malade présente de la fièvre); une proposition composée trouve sa valeur de vérité selon les règles de la logique classique (par exemple, $P \rightarrow Q$ est vrai si P est faux ou Q est vrai).

La théorie des preuves se limite à des schémas simples et catégoriques de raisonnement à savoir le modus ponens (si P et $P \rightarrow Q$, alors Q), le modus tollens (si $\neg Q$ et $P \rightarrow Q$ alors $\neg P$) et le chaînage ($P \rightarrow Q$ et $Q \rightarrow R$ alors $P \rightarrow R$).

Evaluation

Les avantages de ce mode de représentation sont qu'il est complet (toute conséquence logique d'une proposition peut être démontrée), cohérent (on ne démontre que des conséquents logiques) et décidable (on peut déterminer en un nombre fini d'étapes si une proposition est un théorème).

Il convient cependant de mettre en avant certaines limites. Le pouvoir

d'expression est fort restreint du fait que les symboles ne sont pas analysables (il n'y a pas de variable). De plus, la sémantique est rudimentaire (vrai/faux).

12. Logique des prédicats du premier ordre

Formalisme

Le langage contient en plus essentiellement des variables et des quantificateurs. On définit aussi:

- .des termes qui sont -soit une constante ou une variable
-soit une expression du type $F(t_1, t_2, \dots, t_n)$
où les $t_i: 1..n$ sont des termes et F une suite de symboles

.des formules atomiques ou littéraux de la forme $P(t_1, t_2, \dots, t_n)$
où P est une suite de symboles de prédicat.

Pour que les formules soient traitables en machine, elles subissent des transformations pour arriver à une forme clausale telle que: $A_1 \wedge A_2 \wedge \dots \wedge A_m \rightarrow B_1 \vee B_2 \vee \dots \vee B_n$ où les A_i et B_i sont des littéraux toutes les variables étant supposées quantifiées universellement.

Pour tirer des déductions valides du calcul des prédicats, il faut une théorie du premier ordre. Celle-ci se base sur le principe de Robinson permettant, à partir de deux clauses, d'en dériver une troisième appelée résolvente. Le processus se déroule en trois étapes: chercher la paire de littéraux complémentaires, déterminer s'ils s'unifient (s'ils ont la même forme) et effacer cette paire pour déduire la clause résolvente. Nous ne rentrerons pas dans les détails (voir [ROBINSON 65]) mais le type d'inférence étant plus général, les formules dérivables par modus ponens, modus tollens, généralisation et spécialisation le sont aussi par la règle de résolution. L'application de ce principe permet la mise au point d'un algorithme de preuve par réfutation. Signalons dès à présent que le calcul des prédicats du premier ordre et l'algorithme de résolution sont à la base du langage Prolog.

Evaluation

Ce formalisme flanqué du principe de résolution permet la modularité des connaissances, ce qui facilite les modifications. On dispose d'un raisonnement formel cohérent et complet mais indécidable car si on soumet à la procédure de preuve un non-théorème, elle risque de boucler indéfiniment. On peut par ailleurs procéder à des vérifications de cohérence entre fragments de connaissance.

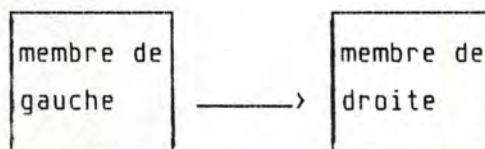
Malheureusement, l'implémentation d'une telle technique de représentation des connaissances est inefficace en temps d'exécution et en en place mémoire. De plus, ce formalisme est inadéquat pour des objets ou relations complexes, qui seront avantageusement représentés par les réseaux sémantiques et les frames que nous étudierons (partie 3.:objets structurés). Enfin, malgré l'emploi des variables, les voies d'expression sont limitées par le manque total de nuance, les deux valeurs de vérité restant vrai et faux.

2.Systèmes de production

Formalisme

Les systèmes de production (SP) comportent un ensemble de règles de production et un mécanisme d'inférence qui travaille selon une base de faits et des objectifs.

Il s'agit de règles de réécriture asymétrique de la forme:



De multiples variantes sont possibles selon les formes que peuvent prendre le premier (MG) et le second (MD) membre des règles. Il ne faut pas voir ces dernières comme une application stricte d'un "si condition alors action" mais plutôt comme le passage d'une situation particulière décrite par une suite de symboles à une nouvelle situation. Dans les systèmes experts, le MG et le MD sont souvent de simples propositions

Logiques qui se présentent sous la forme:

Si $A \wedge B \wedge \dots$ Alors $H1 \wedge H2 \wedge \dots$

La base de faits est un espace de travail qui, dans la situation initiale, contient les données du problème à traiter exprimées sous forme de faits ainsi que l'objectif poursuivi. Elle voit son contenu sans cesse modifié puisqu'elle est constamment mise à jour par l'ajout de nouveaux faits déduits et l'effacement des états antérieurs. L'application des règles de production est donc destructive.

Le mécanisme d'inférence est un cycle reconnaissance/action qui se présente comme suit (fig.1):

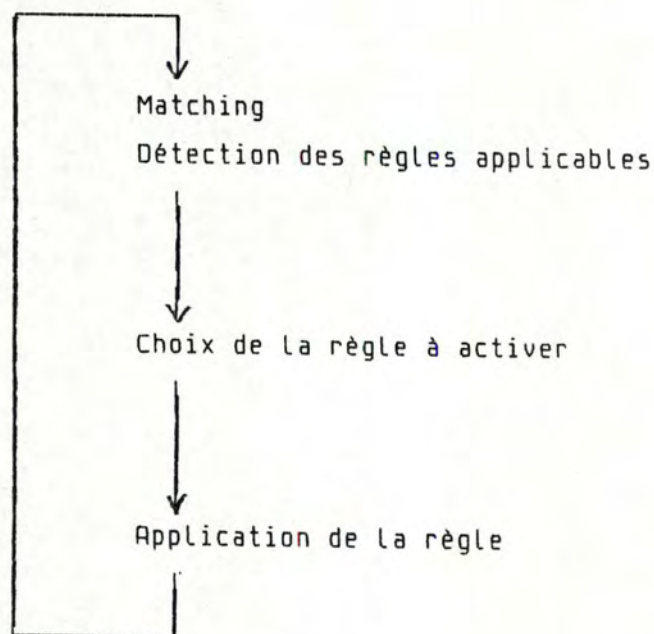


figure 1: cycle reconnaissance/action

Le moteur permet de démontrer un but à partir des faits et des règles. La détermination des règles pertinentes se fait par "pattern matching" ou unification restreinte (pas de variable).

La seconde phase est la résolution de conflit entre plusieurs règles applicables. Cette tâche, qui incombe à l'agenda implique une stratégie de contrôle qui peut être très variée: prendre la première règle trouvée, appliquer un système de priorités dynamiques qui peut être exprimé au moyen de métarègles. Dans ce deuxième cas, de nombreuses possibilités existent (prendre la règle la plus spécifique, celle qui permet de se rapprocher le plus de l'objectif,...).

Enfin, l'application de la règle choisie entraîne la mise à jour de la mémoire de travail et fait avancer les déductions.

La direction de la recherche peut se faire selon deux modes: chaînage arrière ou chaînage avant. Dans un système à chaînage arrière, on part des buts pour atteindre les faits élémentaires. Le contraire se produit dans le chaînage avant. Dans le cas du processus diagnostique en médecine, cela revient à partir des hypothèses diagnostiques jusqu'aux signes cliniques dans le premier cas ou à partir des symptômes et, par déductions successives, arriver aux diagnostics.

En conclusion, on peut donc décrire un système de production selon le schéma suivant (fig.2):

BASE DE CONNAISSANCES

MEMOIRE DE TRAVAIL

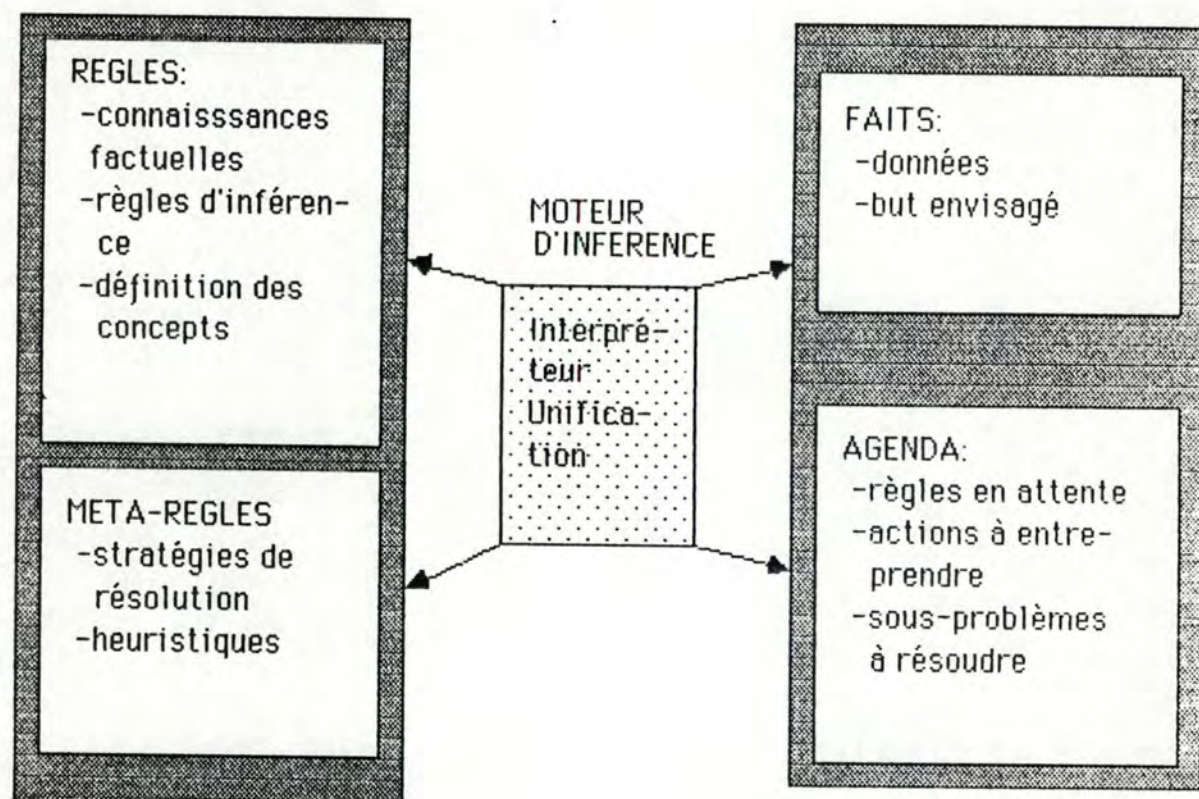


figure 2: schéma général d'un système de règles de production

Evaluation

Les systèmes de production ont l'avantage de rendre la connaissance modulaire: chaque unité d'information est donnée en vrac, comme dans un dictionnaire. Cela correspond à la façon dont l'expert exprime naturellement sa connaissance et il n'est donc pas étonnant que cette technique est utilisée dans l'écrasante majorité des SE. Les effets de bord de la modularité sont une meilleure lisibilité et une modifiabilité plus facile.

Ce mode de représentation est aussi très homogène, ce qui favorise le contrôle de cohérence d'erreurs ainsi que l'auto-explication des déductions, facteurs très importants dans la mise au point d'un système expert.

D'autre part, par rapport à l'ensemble calcul des prédicats-algorithme de résolution, les performances (temps d'exécution/place mémoire) sont meilleures principalement du fait de l'absence de variable.

Enfin, le schéma d'inférence, étant purement formel, est facilement automatisable.

Le principal désavantage, comme on l'a vu, commun aux formalismes basés sur la logique, reste la pauvreté d'expression. Il est souvent difficile de mettre les connaissances d'un expert exclusivement sous forme de règles. Le raisonnement est avantageusement représenté par les systèmes de production. Par contre, il n'est pas naturel d'utiliser cette technique pour représenter des connaissances telles que des relations entre différents objets d'un domaine (les classer selon une relation d'appartenance, par exemple). De ce point de vue, les objets structurés, développés ci-dessous, apportent une plus grande faculté d'expression.

3. Objets structurés

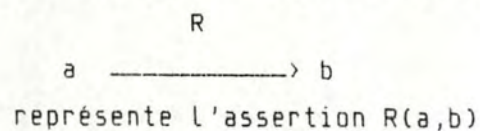
Cette catégorie de modèle, influencés par des études psychologiques, s'est attachée à représenter l'univers dans lequel se situent les problèmes. Contrairement aux techniques précédentes, l'accent est mis sur la représentation de la connaissance plutôt que celle du raisonnement.

Sous la dénomination "objets structurés" sont repris les réseaux sémantiques et les frames.

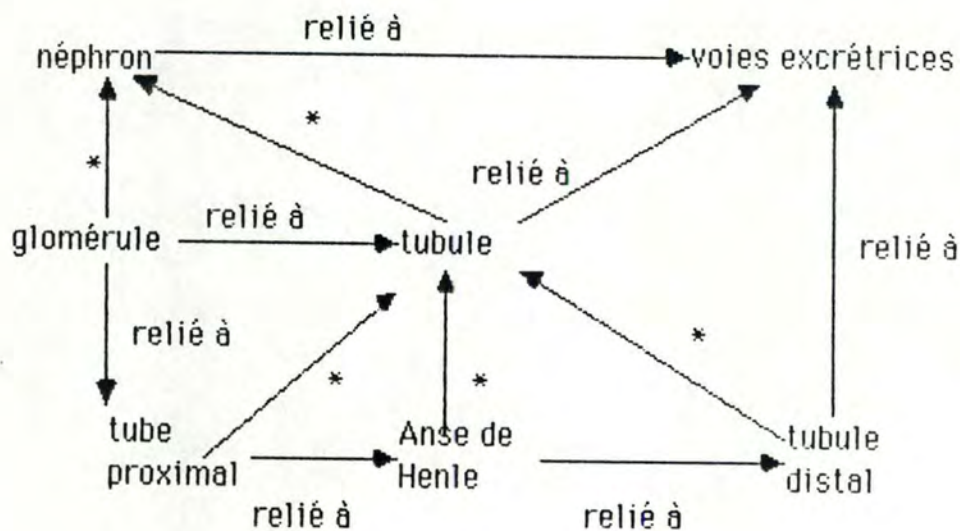
1. Réseaux sémantiques

Formalisme

Un réseau sémantique est un graphe dont les noeuds représentent des concepts, des individus, des situations, des actions et dont les arcs orientés sont des instances de relations binaires. Schématiquement:



C'est ainsi qu'une connaissance anatomique peut se traduire par le graphe suivant [PATIL 81] (fig.3):



N.B.: * est mis pour: "fait partie de"

figure 3: Connaissance anatomique

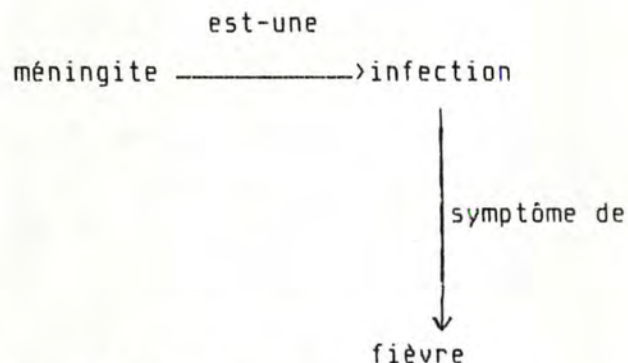
Evaluation

Comme le montre l'exemple, ce formalisme convient particulièrement pour la représentation de concepts fortement interconnectés et interdépendants. Dans ce cas, des regroupements et classifications sont aisément réalisés.

De plus, les relations entre les entités sont explicitées et donnent donc une plus grande faculté d'expression que dans les techniques précédentes, particulièrement pour la description d'objets complexes.

Par ailleurs, il est possible d'avoir un accès très rapide à l'ensemble des informations qui se rapportent à un concept.

Enfin, une particularité remarquable des réseaux sémantiques est l'héritage de propriétés par la relation privilégiée "est-un" ("is-a"). C'est ainsi que si on a, par exemple,



le concept "méningite" hérite de toutes les propriétés du concept "infection" grâce à la relation "est-une" (relation de sous-classe); cela signifie dans cet exemple que fièvre est un symptôme de méningite.

Le principal désavantage des réseaux sémantiques par rapport aux formalismes vus plus haut est qu'il n'existe pas de technique bien définie pour les inférences, ce qui rend la représentation du raisonnement fort complexe. Il n'y a donc pas de vérification formelle possible et les inférences ne sont pas garanties correctes.

Une autre limitation provenant de cette absence de formalisme universel est que la sémantique doit être intégrée dans la structure de contrôle qui varie selon l'utilisation que l'on fait du réseau.

En conclusion, on peut dire que les réseaux sémantiques sont une structure descriptive puissante mais purement passive dont la sémantique est définie de manière floue.

Frames

Formalisme

Cette technique, mise au point par Minsky [MINSKY 75], reprend le principe des réseaux sémantiques mais ajoute une structure plus claire et rigide définissant une sémantique basée sur la notion de classe. Un frame (prototype) est une classe d'objets représentée par l'ensemble des propriétés communes partagées par tout objet de cette classe.

La structure d'un frame est en fait une liste de paires (attribut,valeur) où il y a une partie fixe commune à tous les éléments de la classe et une partie variable différente pour chaque instance. Les attributs représentent l'ensemble des caractéristiques attachées aux objets d'une classe. Ils peuvent être de nature très variée: propriété, relation (avec éventuellement héritage de propriétés si relation est-un, comme dans les réseaux sémantiques), procédures qui définissent "ce que le frame sait faire" ou "ce qu'on peut lui faire faire". Il s'agit dans ce dernier cas d'une sorte de raisonnement local limité. On peut en outre représenter toutes sortes de techniques au sein du frame, comme un prédicat ou une règle de production.

On peut donc remarquer que les frames sont un mode de représentation à la fois déclaratif comme les réseaux sémantiques mais aussi procédural: on parle d'attachement procédural.

Evaluation

Les frames sont, comme les réseaux sémantiques, un puissant formalisme pour représenter des connaissances complexes. On retrouve l'héritage de propriétés (ici d'attribut en général) entre frames qui est systématique sauf mention contraire (pour les exceptions):

En plus, il est possible, au moyen des attributs-procédures d'orienter le contrôle des déductions d'un frame à l'autre. C'est pourquoi, on peut affirmer que les frames offrent une représentation adéquate de la manière dont, au cours de la consultation, le médecin focalise son diagnostic. Ainsi, en l'écoutant, le médecin commence par se dire qu'il s'agit d'un problème digestif; dans le cadre des problèmes digestifs, il s'oriente vers la sphère hépatique; parmi les maladies hépatiques, il envisage une cirrhose; parmi les cirrhoses, une cirrhose post-hépatite, etc...

Par ce procédé, on a la possibilité de faire un très grand nombre de diagnostics, en évaluant jamais qu'un petit nombre d'hypothèses diagnostiques à la fois.

Le désavantage de cet attachement procédural est la difficulté de valider un raisonnement, difficulté que l'on retrouve dans la programmation classique.

Enfin, comme dans les réseaux, la sémantique est fonction du mode d'utilisation des frames et est donc non explicite.

3.2.4)SYNTHESE

Il est difficile de comparer les différents formalismes étudiés sur base de leur capacité de représenter les connaissances dans un SE. Buchanan [BUCHANAN 82] souligne trois critères à prendre en compte:

- l'extensibilité: les structures de données doivent être assez souples pour permettre des extensions de la base de connaissance sans nécessiter de profonds remaniements;
- la simplicité: la représentation et l'expression de la connaissance doivent être simples et accessibles pour un non informaticien;
- la connaissance doit être explicite: ceci est un élément important pour la recherche des erreurs et l'expression des explications.

Les formalismes basés sur la logique mathématique et surtout les systèmes de production rencontrent ces trois objectifs. Par contre, les réseaux sémantiques et les frames ont des structures moins formalisées et donc plus complexes et expriment la sémantique en fonction du mode d'utilisation des connaissances, ce qui rend difficile la validation et l'auto-explication du raisonnement. Ce dernier point est surtout vrai pour les frames en raison de l'attachement procédural qui complique l'interprétation. En effet, les frames empêchent la neutralité de la représentation vis-à-vis de l'utilisation ultérieure qu'en fera les mécanismes d'inférence.

D'un autre côté, les réseaux sémantiques et plus encore les frames apportent une plus grande puissance d'expression des objets complexes. Ils permettent de classifier les objets, les concepts et offrent l'héritage de propriétés entre des objets reliés par la relation "est-un". Ils répondent donc mieux à la diversité des connaissances que nous avons montrée plus haut (voir 3.2.2:typologie des connaissances).

On serait tenté d'affirmer que les frames sont indiqués pour décrire des objets statiques tandis que les règles de production conviennent parfaitement pour représenter la dynamique d'un raisonnement déductif. Mais il faut éviter d'opposer les formalismes dans la mesure où, on s'en est rendu compte, ils sont complémentaires. Il est d'ailleurs parfaitement possible de mettre comme attribut d'un frame une règle de production.

Il n'existe pas de technique de représentation idéale et une seule d'entre elles ne peut répondre à la complexité des connaissances et du raisonnement médicaux. Il sera intéressant de voir l'application de ces techniques dans la réalisation de systèmes experts médicaux; c'est ce que nous ferons dans la partie qui suit.

3.3) EXEMPLES DE SYSTEMES EXPERTS MEDICAUX

Depuis le début des recherches sur les SE, la médecine a été un champ d'application largement utilisé. De nombreuses réalisations ont vu le jour dont certaines sont actuellement opérationnelles. Nous reprenons succinctement les principales d'entre elles en mettant l'accent sur les différentes techniques de représentation des connaissances et de modélisation du raisonnement.

3.3.1) MYCIN [SHORTLIFFE 76]

Objet

Ce SE, un des premiers et le plus connu, est destiné à conseiller les médecins sur le choix d'un antibiotique adapté au traitement de patients ayant une infection bactérienne avant que le germe responsable ne soit identifié. Il s'agit d'un problème complexe mais bien délimité.

Le raisonnement de MYCIN se fait en l'absence d'information complète et est basé sur un ensemble de connaissances exprimées sous forme de règles de production. Il est capable d'expliquer son raisonnement et possède un modèle de sa connaissance (il sait ce qu'il sait).

Son intérêt réside dans l'emploi de techniques générales (indépendantes de l'application) et dans la qualité des résultats qu'il fournit.

MYCIN, bien que testé en milieu clinique, reste un projet expérimental mais représente une référence dans le domaine des systèmes experts.

Description générale

Le système se compose essentiellement d'une base de connaissance qui est un ensemble de règles et de quatre programmes principaux qui coopèrent (fig.4):



figure 4: Schéma de principe de Mycin

Représentation des connaissances

La base de connaissances comporte environ 200 règles de production dont les antécédents sont normalisés sous forme de quadruplets: <prédicat> <objet> <attribut> <valeur>.

Exemple:

MYCIN: règle 85

SI 1° Le site de la culture est le sang,
et SI 2° L'organisme est à Gram négatif,
et SI 3° L'organisme est de forme bâtonnet,
et SI 4° Le patient est un hôte à risque

ALORS il est probable (0.6) que l'organisme est le
pseudomonas aeruginosa

Dans la proposition "le site de la culture est le sang", par exemple, on a donc respectivement <est> <culture> <site> <sang>. Il existe des règles pour le diagnostic et d'autres pour les prescriptions.

Raisonnement approché

L'exemple montre l'emploi d'un coefficient (0.6), le coefficient de la règle, qui exprime le degré de certitude exprimé par l'expert à propos de la conclusion annoncée. Les faits eux-mêmes sont représentés par des triplets valués par un coefficient de vraisemblance compris entre -1 et 1, comme par exemple: (identité de l'organisme: E.Coli 0.7). Il est important de noter que ces coefficients ne sont pas des probabilités mais des mesures subjectives qui permettent de modéliser le raisonnement approché.

Mécanisme d'inférence

Le raisonnement se fait en chaînage arrière en partant des diagnostics possibles (micro-organismes) vers les symptômes par la construction d'un arbre ET-OU. Un but étant donné, le système considère toutes les règles dont les conséquents portent ce but. L'antécédent de la règle est alors évalué et le conséquent se voit affecté d'un coefficient de vraisemblance par calcul. Si le système ne parvient à aucune conclusion, il demande de nouvelles informations relatives au patient. MYCIN procède donc par énumération exhaustive à rebours, ce qui le rapproche d'un démonstrateur de théorèmes travaillant dans un espace de recherche relativement réduit.

Métarègles

Le système a accès à sa propre connaissance par les métarègles qui sont des règles qui permettent de piloter sa recherche.

Exemple: METAREGLE 2:

SI 1° le patient est un hôte à risque,
et SI 2° il existe des règles qui mentionnent des
pseudomonas dans une prémisse,
et SI 3° il existe des règles qui mentionnent des
klebsiellas dans une prémisse,
ALORS il est probable (0.4) qu'il faille utiliser les
premières avant les secondes.

Les métarègles permettent ainsi de modéliser les heuristiques et la structure de contrôle indépendamment des autres connaissances.

Explication et dialogue en langage naturelle

Pour les médecins, il est capital de fournir l'explication du raisonnement, ce qui est possible car le système garde la trace de son raisonnement et des règles utilisées.

L'interface qui permet un dialogue en un langage proche du langage naturel est effectuée par un interprète élémentaire dont la tâche est facilitée par la forme standard de la connaissance.

3.3.2) SPHINX [FIESCHI 86]

Objet

Le but de SPHINX est de constituer une aide à la décision dans le domaine du diagnostic et du traitement. Le système n'est pas dédié à une application particulière mais a été utilisé pour le diagnostic différentiel des ictères et le traitement du diabète. Un effort tout particulier a été fait pour se rapprocher de la démarche du médecin en simulant une consultation. Les spécifications de l'outil sont de procurer un avis comparable à celui d'un expert, par un dialogue dans un langage proche du langage naturel, de fournir des possibilités d'explication et de permettre une évolution du système. Ces objectifs cadrent parfaitement aux possibilités que peut offrir un système expert.

SPHINX est destiné à servir en pratique courante; il devrait être prochainement accessible aux généralistes via le Minitel français pour leur offrir son expertise dans le traitement du diabète.

Description générale (fig.5)

L'utilisateur entre en communication avec SPHINX grâce au module MEDIUM qui gère le dialogue en manipulant des connaissances relatives au discours médical. Le module d'aide à la décision proprement dit, appelé EXPERT, recherche les signes cliniques et résultats d'examens complémentaires pour arriver à produire ses conclusions (diagnostics, traitements).

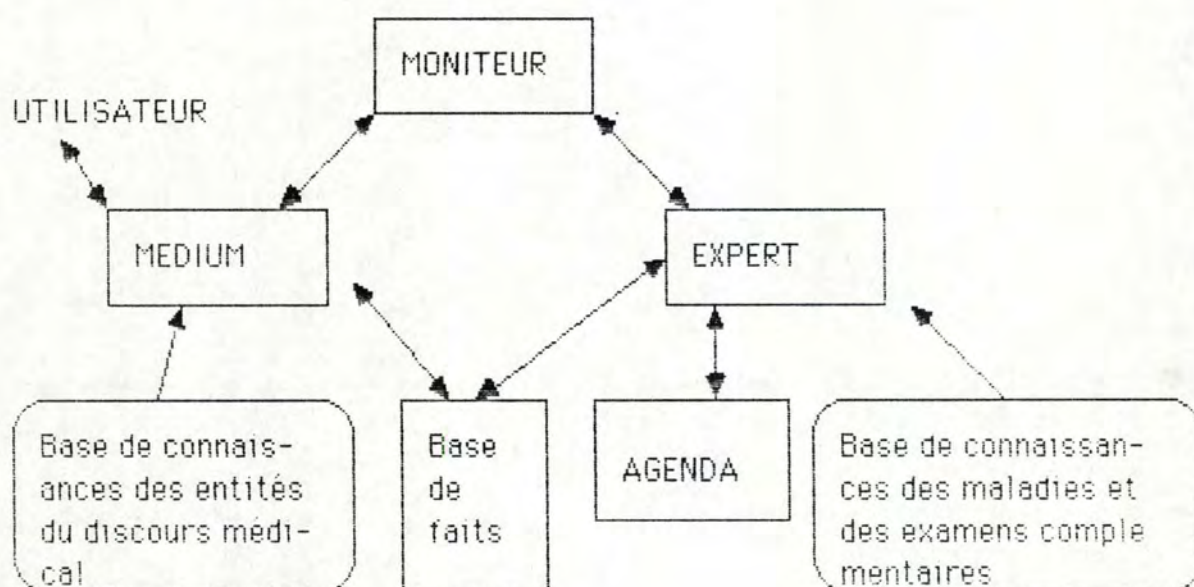


figure 5: Présentation générale de Sphinx

Représentation des connaissances

Comme le montre la figure ci-dessus, la base de connaissance est scindée en deux parties.

La première comporte les entités du discours médical qui sont des concepts (symptôme, signe clinique,...) comportant des variables sémantiques prenant leurs valeurs dans des catégories sémantiques. Par exemple, on aura (dans l'ordre) le symptôme "vôissement" dont la variable "fréquence" prend la valeur "souvent". La similitude avec le

triplet <objet> <attribut> <valeur> est évidente. L'organisation de cette connaissance s'inspire des frames de Minsky, les entités étant les frames, les attributs les slots et l'attachement procédural représenté par des routines qui définissent des conditions de réalisation ou des traductions d'informations quantitatives (température = 36.5°) en valeur sémantique (température normale).

La seconde base de connaissance est celle du module EXPERT. La représentation est sous forme de règles de production. Il existe quatre formes de règles:

1° règles d'évocation: elles sont utilisées en chaînage avant et servent à la mise à jour de l'agenda en lui fournissant les diagnostics que le système doit tenter d'établir en fonction du contenu de la base de faits.

2° règles de connaissance: elles contiennent les informations relatives aux diagnostics et aux traitements et travaillent par chaînage arrière, ce qui permet l'évaluation exhaustive des diagnostics retenus par l'évocation et l'expression de la démarche pas-à-pas du médecin.

3° règles d'interprétation d'examens: par chaînage avant, elles réalisent la prise en compte de tous les éléments que révèle l'examen et non une interprétation limitée dans le cadre du but que le système cherche à démontrer.

4° règles de résolution de conflit: également utilisées en chaînage avant, elles constituent des métarègles visant à optimiser la recherche de solutions.

Modélisation du raisonnement

Le raisonnement médical est remarquablement modélisé. L'élaboration d'hypothèses se fait par un raisonnement approché utilisant la logique floue au sein de règles d'évocation qui sont activées à partir de symptômes d'appel répertoriés dans la base de faits. Cette évocation a pour résultat de définir des contextes qui sont en fait un ensemble d'hypothèses diagnostiques qui sont communiquées à l'agenda.

Ensuite, pour chaque hypothèse, le module EXPERT travaille en chaînage arrière pour tenter d'établir, au moyen d'un raisonnement certain, le diagnostic considéré. Si des informations manquent, MEDIUM se charge de

les demander à l'utilisateur.

Explication

L'état du raisonnement constitue une connaissance spécifique accessible à tout moment à l'utilisateur, ce qui permet au système de fournir des explications sur ses conclusions.

3.3.3) INTERNIST [MILLER 82]

Objet

Ce système, conçu par un chercheur informaticien et par un spécialiste en médecine interne, est un programme d'aide au diagnostic qui couvre presque l'entièreté de la médecine interne. Il est basé sur une étude minutieuse du raisonnement médical qu'il tente d'automatiser.

INTERNIST reste pour le moment, au yeux de ses initiateurs, un outil de recherche. Bien que développé dans un environnement clinique et testé sur des dossiers de malades réels, il n'est pas encore utilisé comme consultant sur des cas réels.

Représentation des connaissances

La base de connaissances associe à chaque diagnostic possible Di un ensemble de manifestations (mj). Une manifestation est un signe, un symptôme, un résultat de laboratoire ou un autre diagnostic qui peut être associé. Le système a une structure hiérarchique reprenant les différents chapitres de la pathologie tels les maladies du foie, du coeur,...

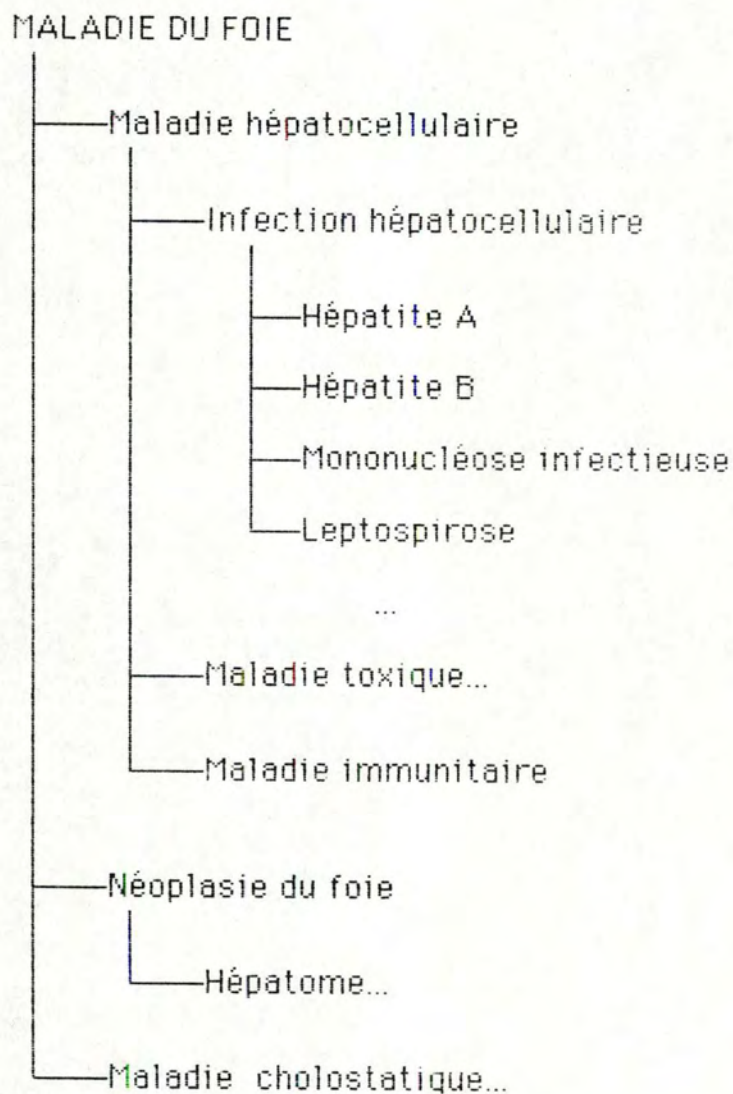
Exemple:

figure 6:Hiérarchies

Ces hiérarchies sont réalisées par la technique des frames de Minsky. Elles permettent de contrôler la prolifération des hypothèses actives pendant la consultation. Un diagnostic est dit actif si au moins une de ses manifestations a été observée.

Il existe trois relations importantes pour utiliser la connaissance médicale de base. La relation EVOQUE entre une manifestation et une maladie, la relation FORM-OF qui indique qu'une maladie est une forme d'une autre et la relation MANIFEST qui est la relation inverse de EVOQUE entre la maladie et sa manifestation.

Cette structure a l'avantage de fournir un certain degré de focalisation qui permet d'éviter l'explosion combinatoire dans la recherche.

Modélisation du raisonnement

Au départ, on présente à INTERNIST des données concernant un patient (telle manifestation est présente, telle autre absente). Pour chaque manifestation, le système engendre la liste de toutes les causes, ou hypothèses diagnostiques, possibles. Ensuite, pour chaque hypothèse, il dresse quatre listes: celle des manifestations qu'elle explique, celle des manifestations qu'elle n'explique pas, celle des manifestations qui devraient normalement être présentes et qui manquent au tableau, celle des manifestations à rechercher soit par des questions soit par des examens à faire.

INTERNIST connaît les profils de maladie qui sont une liste des manifestations dotées, chacune, de plusieurs index: force avec laquelle le signe évoque la maladie, fréquence avec laquelle le signe est rencontré dans la maladie, type (symptôme, signe clinique, résultat d'examen) et degré de gravité intrinsèque de la manifestation. A partir de ces index, INTERNIST calcule un score pour chaque hypothèse diagnostique et donne la liste des hypothèses par ordre de scores décroissants.

Ensuite, il crée une zone de problème, incluant l'hypothèse ayant le score le plus élevé, et toute les hypothèses entrant en compétition avec elle et qui rentrent dans le même cadre. Par une série de questions, les hypothèses sont départagées en entraînant une réévaluation des scores (diagnostic différentiel). Lorsque le score d'une hypothèse dépasse un certain seuil, l'hypothèse est retenue et INTERNIST crée une nouvelle zone de problème pour tenter d'expliquer d'autres manifestations.

La caractéristique principale du fonctionnement de ce système est de centrer l'attention sur une zone de problème sans devoir tenir compte de

toutes les données à la fois. C'est exactement la stratégie adoptée par le médecin dans son raisonnement.

L'évaluation du système a mis en évidence la nécessité d'intégrer des possibilités de raisonnement à partir d'éléments anatomiques, de données temporelles, et de proposer des diagnostics différentiels pour des patients présentant plusieurs pathologies à la fois. Ces possibilités sont disponibles dans la nouvelle version du système: CADUCEUS [POPLE 82].

3.3.4) CASNET [WEISS 78]

Objet

Ce SE a été appliqué à l'aide au diagnostic et à la thérapeutique des glaucomes. Cette pathologie a la particularité, encore rare en médecine, que les séquences d'états physiopathologiques qui sous-tendent l'évolution de la maladie sont bien connues. Les performances de ce SE, testé par un groupe d'ophtalmologistes américains, sont remarquables puisque 95% de résultats satisfaisants ont été observés.

Représentation des connaissances

La connaissance médicale est représentée par un réseau sémantique d'états physiologiques et physiopathologiques reliés par des relations de causalité (Causal ASSociation NETwork). L'accent est mis sur la modélisation de la compréhension des mécanismes et de l'évolution de la maladie.

Modélisation du raisonnement

Contrairement aux autres systèmes étudiés jusqu'à présent, le système ne travaille pas à partir d'hypothèses diagnostiques. La première phase est la saisie des informations concernant le patient. Ceci se fait donc en bloc sans aucune recherche de diagnostic.

Ensuite, d'après ces observations, le système figure sur le réseau causal l'état physiopathologique actuel du patient, en confirmant certains états et en infirmant d'autres. Cette deuxième étape permet déjà d'émettre un pronostic sur l'évolution et de suivre la progression

des événements d'une consultation à l'autre.

La phase de diagnostic n'est enclenchée qu'alors et définit des états maladies qui sont d'un niveau d'abstraction supérieur.

Enfin, le système propose un traitement en produisant des états thérapeutiques reliés entre eux par des contraintes (interactions médicamenteuses, toxicité,...).

Il est important de constater le caractère très particulier de la modélisation du raisonnement de ce SE. En fait, les auteurs ont choisi cette application parce qu'elle est bien définie, bien circonscrite et où les mécanismes de la maladie sont bien connus. L'utilisation d'un schéma causal convenait donc parfaitement bien mais cette technique s'avère difficile à transposer dans d'autres domaines.

PARTIE II: APPLICATION

4) LE PROJET

4.1) DESCRIPTION GENERALE

Le projet s'inscrit dans le cadre des applications informatiques développées à la Clinique St Jean à Bruxelles. Celle-ci dispose d'un système hospitalier intégré avec environ 200 terminaux répartis dans plusieurs cliniques. Un logiciel, mis au point en collaboration avec des hôpitaux universitaires hollandais, gère de multiples applications, tant médicales qu'administratives. Une équipe d'environ quinze informaticiens est chargée de l'exploitation du système et du développement de nouvelles applications. L'intérêt des utilisateurs va croissant, notamment parmi les médecins.

L'éclosion de systèmes d'aide à la décision en médecine permet d'imaginer que de tels outils seront, dans un avenir plus ou moins proche, utilisés en pratique courante dans les hôpitaux. Afin de s'y préparer, l'idée de mettre au point un système expérimental et en dimension réduite s'est avérée intéressante dans la mesure où cela nécessitait une approche concrète et constructive du sujet. Dès le début, l'accent a été mis sur l'emploi de techniques propres à ce que l'on appelle l'intelligence artificielle (en particulier les systèmes experts). Il est donc important de constater que ce projet relève plus de la mise au point d'un prototype expérimental que de l'informatisation d'une tâche à partir de besoins spécifiques. Il n'est évidemment pas du ressort d'un hôpital comme la Clinique St Jean de réaliser un système expert en vraie grandeur quand on connaît l'investissement que cela représente en temps de travail et en matériel. C'est la raison pour laquelle ce projet a été fixé dans le cadre de ce mémoire et avec des moyens compatibles aux possibilités et aux objectifs. C'est ainsi que la durée du travail se limitait à un an à temps partiel et qu'un PC Olivetti M24 avec 640k de M.C. et disque dur a été mis à disposition. Le logiciel Turbo Prolog a été dès le départ proposé pour des raisons de facilité et d'économie.

Le lieu de développement de ce projet expérimental est le service des soins intensifs où l'équipe des médecins et des infirmières est habituée à utiliser des terminaux, en particulier dans la communication de données du laboratoire. De plus, l'implication de l'informatique dans ce département est essentielle (monitoring, par exemple) et l'intérêt est grand. Le système expérimental est appelé à servir d'aide dans la mise au point d'un patient comateux en urgence. C'est donc au service des urgences qu'est destiné cet outil même si il a été développé en collaboration avec l'équipe des soins intensifs dont les médecins sont par ailleurs responsables de l'organisation des urgences.

Dans le domaine de l'aide à la décision, il est apparu que la mise au point d'un patient comateux nécessitait des prises de décision et un diagnostic rapides parfois en présence de peu d'éléments d'information. Le choix de actes à poser et la démarche diagnostique se base sur des critères systématisables bien que le nombre d'hypothèses soit élevé. Le problème est trop complexe pour être décrit par un algorithme ou même par un arbre de décision reste suffisamment délimité pour être appréhendé dans son ensemble.

Les besoins en information se situe essentiellement dans l'opportunité de poser certains gestes thérapeutiques et diagnostiques en fonction de données incomplètes et d'envisager les multiples causes de coma en recherchant les signes importants. Pour des médecins ou des infirmières débutants, il est difficile de mener une démarche cohérente et efficace face à un patient comateux. Il serait donc utile de mettre à leur disposition un système d'aide qui puisse refléter à la fois la connaissance médicale dans ce domaine et l'expérience de médecins réanimateurs.

4.2)OBJECTIFS

Il s'agit de produire un système expérimental d'aide à la décision et au diagnostic pour les médecins et infirmières débutants qui puisse:

- 1°Susciter l'intérêt du corps médical pour le domaine de l'aide à la décision par ordinateur.
- 2°Préparer l'avenir quant à l'utilité, les possibilités mais aussi les difficultés et les limites de tels outils.
- 3°Systématiser la mise au point d'un patient comateux par la mise en évidence des critères de décision et de diagnostic.

5)LE SUJET: MISE AU POINT D'UN PATIENT COMATEUX

5.1)INTRODUCTION

Le coma n'est pas une maladie en soi mais représente l'expression d'une pathologie sous-jacente. Parfois, le diagnostic est évident, par exemple quand un individu sain reste inconscient après un traumatisme crânien. Cependant, trop souvent, un patient comateux est amené en urgence sans que l'on puisse disposer

de suffisamment d'information pour orienter le diagnostic. Dans ce cas, le personnel est amené à poser des gestes qui peuvent se révéler d'une importance vitale pour la survie du patient. Le plus souvent, les premières mesures ont une portée thérapeutique visant la conservation des fonctions vitales. Ce n'est qu'ensuite que commencera la démarche diagnostique sur base des données de l'interrogatoire des personnes accompagnantes, de l'examen clinique et des examens complémentaires.

Il existe deux types de connaissances qui doivent être explicitées dans ce chapitre: celles relatives à l'expérience pratique du médecin qui permet de définir une démarche (connaissance empirique) et celles qui décrivent de manière purement formelle le coma comme entité pathologique (connaissance scientifique). Même si, en pratique, elles sont liées, la description faite dans la partie qui suit (partie 4.2: La démarche) s'attache plus au premier type de connaissance tandis que les parties restantes du chapitre (parties 4.3 à 4.7) traitent de la connaissance scientifique.

5.2) LA DEMARCHE

Il est difficile de systématiser la démarche du médecin face à un patient comateux. En effet, il n'existe pas de solution algorithmique permettant d'explicitier, étape par étape, le processus. D'autre part, il existe des différences selon les médecins, selon leur degré d'expérience ou de spécialisation.

Pourtant, il est possible de dégager des points de passage obligés, des fonctions essentielles communément admises. C'est ainsi que l'on peut constater que dans un premier temps, l'accent est mis sur la conservation des fonctions vitales. Un patient en coma est une urgence médicale qui nécessite une vérification de la fonction cardiaque, de la respiration, de l'état du cerveau et de la température. A ce stade, les gestes ont plus une portée thérapeutique qu'une portée diagnostique même si une foule de données utiles sont récoltées pendant cette première phase.

Le médecin dispose essentiellement de quatre sources d'information: interrogatoire de la famille, étude du dossier antérieur du malade, examen clinique et examens complémentaires. Il est important de distinguer ces quatre sources pour plusieurs raisons. Elles se différencient d'abord d'après la nature de l'information qu'elles apportent: l'interrogatoire de la famille permettront de préciser les circonstances d'apparition du coma, l'étude du dossier fournira les antécédents du malade tandis que l'examen clinique et les examens complémentaires objectiveront l'état actuel du malade. Parfois, la famille ou un dossier antérieur fait défaut. D'autre part, l'examen clinique précède les examens complémentaires qui peuvent selon les cas fournir des résultats après un certain délai. Enfin, le poids des différentes sources d'information dans la prise de décision et le diagnostic est sensiblement différent. Le plus souvent, la famille et le dossier donnent des indications qui permettent d'orienter la recherche tandis que l'examen clinique et les examens complémentaires doivent apporter des éléments permettant des décisions et des conclusions.

Il est important de noter que deux aspects fondamentaux se dégagent de la démarche: un aspect procédural et un aspect statique. Le premier traduit la séquence de gestes à poser face à l'urgence d'un coma. L'ordre a de

L'importance, fonction de la priorité relative de ce qu'il convient de faire et du déroulement logique des opérations (examen clinique avant radiographie, par exemple). L'aspect statique répond à la question suivante: à partir d'un ensemble donné d'informations, que peut-on déduire, quel diagnostic peut-on poser? Diagnostic est pris ici au sens large, exprimant des informations déduites à partir d'autres. Cet aspect statique n'est pas figé à un moment unique, il ne s'agit pas de faire un diagnostic une fois pour toutes à partir de données complètes. Au contraire, à chaque étape, en fonction d'une situation donnée, le médecin fait le point de ses déductions pour prendre une décision ou poser un diagnostic.

5.3) LES PREMIERS GESTES

Le coma étant une urgence médicale, il importe de poser des gestes qui peuvent s'inscrire dans un processus dont les étapes sont les suivantes:

1) FONCTIONS VITALES:

A-B-C: airway-breathing-circulation

- * vérifier la liberté des voies aériennes
- * respiration (dépression, hyperventilation, type Kussmaul (*), type Cheyne-Stokes)
- * examen cardiovasculaire (T.A., pouls, auscultation cardiaque)

!----> réanimation cardio-pulmonaire si nécessaire

intubation + respirateur si difficulté de respiration (chute de langue,...) et de toute façon si score de Glasgow ≤ 9 .

température rectale.

2) EXAMEN CLINIQUE

- examen neurologique : signes de latéralisation, signes d'hypertension intracrânienne, mouvements oculaires, pupilles,
- peau: sèche ou transpiration, cyanose
- haleine: odeur d'acétone, d'alcool

3) POSITIONNER LE MALADE:

si suspicion de traumatisme crânien
alors craindre une fracture cervicale et stabiliser le cou
sinon mettre en position latérale de sécurité.

4) EXAMENS URGENTS:

- prise de sang: cofo, urée, créatinine, enzymes hépatiques, glucose, groupe sanguin, ionogramme, tests de coagulation, tests préopératoires, 2 tubes pour la recherche toxicologique.
- gazs artériels (évaluer l'oxygénation, utilité diagnostique dans les intoxications)
- ECG
- tigette DEXTROSTIX.
 - Scanner cérébral
 - si température ou raideur de nuque: ponction lombaire (P.L.)
 - EEG: si suspicion d'encéphalite (coma fébrile sans raideur de nuque)

5) TRAITEMENTS URGENTS:

- si hypoglycémie ou doute
alors 40 ml glucose 50% IV sur 5'
- si suspicion d'overdose (myosis,...): naloxone (NARCAN) 1 amp. 0.4 mg IV (test diagnostic: amélioration en cas d'overdose morphinique ou intoxication à l'alcool)
- si alcoolisme: thiamine (p.ex. BEROLASE) 50-100 mg IV (encéphalopathie de Wernicke)
- si convulsions: phénitoïne 15-20 mg/kg (1000-1500 mg).

6) DETERMINER:

-circonstances d'apparition (hétéroanamnèse):

déterminer: .caractère progressif ou brutal

-antécédants médicaux (cardiovasculaires, tendance suicidaire, toxicomanie, épilepsie, diabète).

7) PROFONDEUR DU COMA:

Elle se mesure à l'aide de l'échelle de GLASGOW qui tient compte de trois paramètres cliniques:

<u>-ouverture des yeux:</u>	spontanée	4
	aux ordres	3
	à la douleur	2
	aucune	1
<u>-réponse motrice:</u>	aux ordres	6
	à la douleur	5
	adaptée	4
	inadaptée	3
	extension	2
	aucune	1
<u>-réponse verbale:</u>	normale	5
	confuse	4
	délirante	3
	inintelligible	2
	aucune	1

Un coma classé ≤ 8 doit être considéré comme grave et faire envisager des manoeuvres de réanimation. Le transfert en soins intensifs est indispensable.

5.4) CAUSES PRINCIPALES DE COMA

La liste qui suit n'est pas exhaustive mais reprend les diagnostics principaux à envisager devant un patient en coma.

1. TRAUMATISMES CRANIENS

- hématome intracrânien, commotion cérébrale

2. INTOXICATIONS

- alcool (coma éthylique)
- médicaments: benzodiazépines, hypnotiques non barbituriques, barbituriques, tricycliques.
- drogues

3. AFFECTIONS NEUROLOGIQUES

- accidents vasculaires cérébraux: thrombose, embolie, hémorragie
- tumeurs et lésions cérébrales extensives

4. AFFECTIONS METABOLIQUES

- dans le cadre du diabète: acidocétose, coma hyperosmolaire, hypoglycémie
- désordres hydro-électriques et acido-basiques
- insuffisance rénale
- insuffisance hépatique
- carence vitaminique (thiamine)

5. INFECTIONS

- méningite.
- encéphalite virale
- abcès cérébral
- septicémies (pneumonies, fièvre typhoïde, malaria,...)

6. COLLAPSUS CARDIO-VASCULAIRE (CHOC)

7.TROUBLES THERMIQUES

-hyperthermie (coup de chaleur), (hypothermie)

8.DIVERS:

-état post-critique (épilepsie)

Les causes les plus fréquentes sont, par ordre décroissant:

1.intoxications	53%
2.affections neurologiques	27%
3.traumatismes	12%
4.affections métaboliques	8%

5.5)LISTES DE SIGNES PAR DIAGNOSTIC

On définit une liste de signes qui peuvent être de différents types: .données d'ordre général [DO] (âge, sexe, médicaments pris régulièrement,...)

.antécédent [AT] (passé médical du patient)

.symptôme [SY] (plaintes, histoire de l'épisode actuel)

.signe clinique [SC] (donnée provenant de l'examen clinique)

.résultat d'examen complémentaire [EC] (prise de sang, radiographie, ...).

Il est important de noter que ces signes sont donnés sans considération de poids dans le diagnostic. Ils peuvent être plus ou moins spécifiques ou sensibles. Leur point commun est d'être potentiellement présent lorsque la patient souffre d'une affection donnée.

NB: les termes marqués de (*) seront explicités dans un glossaire (voir 5.7: GLOSSAIRE)

1°Traumatisme crânien simple

SY: notion de traumatisme

SC: hémorragie auriculaire, hémorragie nasale, plaie du cuir chevelu, hémotympan,

EC: radio de crâne: signe de fracture,

2°Traumatisme crânien avec hématome intracrânien compressif

SY: notion de traumatisme, intervalle lucide entre l'accident et la perte de connaissance

SC: hémorragie auriculaire, hémorragie nasale, plaie du cuir chevelu, hémotympan, signes de latéralisation, signes d'hypertension intracrânienne

EC: scanner: image d'hématome intracrânien

3°Coma éthylique

DO:

AT: patient alcoolique

SY: notion d'excès d'alcool, stigmates d'alcoolisme (angiomes stellaires)

SC: hypothermie, hypotension, stigmates d'alcoolisme (hépatomégalie), haleine: odeur d'alcool

EC: stigmates d'alcoolisme à la biologie, alcoolémie élevée

4°Intoxication médicamenteuse

DO: médicaments potentiellement dangereux
(benzodiazépines, barbituriques, hypnotiques non barbituriques,
tricycliques)

AT: tendance suicidaire

SY: notion d'abus de médicament potentiellement dangereux

SC: coma calme, apyrétique, flasque, sans signe de localisation
neurologique, réflexes O.T. faibles ou abolis, myosis ou mydriase
avec réflexe photomoteur normal, tremblements

EC: médicament dans le sang

5°Intoxication par drogue

DO:

AT: patient drogué

SY: notion de prise de drogue

SC: taces de piqures à l'avant-bras, myosis bilatéral

EC: drogue dans le sang

6°Affection neurologique (Accident vasculaire cérébral, tumeur ou

lésion cérébrale extensive)

DO:

AT: accident vasculaire cérébral, tumeur cérébrale, lésion cérébrale

extensive

SY:

SC: signes de latéralisation, signe de localisation neurologique

EC: EEG (signe d'atteinte neurologique), Scanner (signe d'atteinte neurologique)

7°Hypoglycémie du diabétique

DO: insuline ou médicaments hypoglycémiants

AT: patient diabétique

SY: coma d'apparition brutale (<1h)

SC: pas de déshydratation, transpiration

EC: hypoglycémie

8°Acido-cétose

DO: insuline ou médicaments hypoglycémiants, enfant (souvent)

AT: patient diabétique (le plus souvent de type I)

SY: coma d'apparition lente (>1j), facteur d'acido-cétose (infection, stress, diminution ou arrêt d'insuline ou de médicament hypoglycémiant), signes de cétose (nausées, vomissements)

SC: haleine: odeur d'acétone, déshydratation, dyspnée de Kussmaul (*)

EC: hyperglycémie, acidose métabolique, corps cétoniques fort augmentés, glycosurie, acétonurie

9° Coma hyperosmolaire

DO: adulte (souvent)

AT: patient diabétique (surtout de type II)

SY: coma d'apparition lente (>1j)

SC: déshydratation, signes neurologiques (hémiparésie, signe de Babinski), pas de dyspnée de Kussmaul (*)

EC: hyperglycémie majeure, pas d'acidose métabolique nette, pas d'acétonurie, corps cétoniques normaux ou peu augmentés

10° Désordres hydro-électriques et acido-basiques

EC: augmentation ou diminution du taux de Na⁺, K⁺, Ca⁺⁺, magnésium; hyper- ou hypo-osmolalité; acidose (métabolique, respiratoire), alcalose (métabolique, respiratoire)

11° Insuffisance rénale

AT: patient insuffisant rénal connu

SC: convulsions

EC: urée, créatinine augmentés

12° Insuffisance hépatique

AT: patient insuffisant hépatique connu

EC: enzymes hépatiques perturbés

13° Carence en Vit B1

DO:

AT: patient alcoolique

SY: amélioration en cas d'administration de Vit B1 IV

SC:

EC:

14° Méningite

DO:

AT:

SY: notion de céphalées, de vomissements

SC: fièvre, signes d'infection locale ou générale, raideur de nuque, convulsions

EC: syndrome inflammatoire à la biologie, ponction lombaire (P.L.) positive

15° Encéphalite virale

DO:

AT:

SY: infection virale les jours précédents

SC: température, convulsions, signes neurologiques (hémiplégie, paraplégie, syndrome cérébelleux, paralysie de nerfs crâniens)

EC: biologie compatible avec une infection virale, EEG, P.L.

16° Abscès cérébral

DO:

AT:

SY: céphalées, notion de traumatisme des sinus, infection O.R.L. ou pulmonaire, crise d'épilepsie

SC: fièvre, signes neurologiques, signes d'hypertension intracrânienne (*)

EC: biologie compatible avec une infection bactérienne, EEG, Scanner

17° Septicémies graves

DO:

AT:

SY: frissons

SC: fièvre

EC: hémoculture positive, goutte épaisse (malaria)

18°Choc

DO:

AT:

SY:

SC: pouls filant, tachycardie, hypotension, cyanose

EC:

19°Hyperthermie (coup de chaleur)

DO:

AT:

SY: exposition au soleil

SC: température très élevée, peau sèche

EC:

20°Etat post critique (épilepsie)

DO:

AT: patient épileptique

SY: coma de durée < 30', notion de crise d'épilepsie

SC:

EC: EEG

5.6) CRITERES DE DIAGNOSTIC

Les entités diagnostiques sont reprises à titre individuel pour en spécifier les critères. Il existe, pour un diagnostic, quatre états possibles: diagnostic évoqué (EV), suspecté (SU), prouvé (PR), ou exclu (E).

DIAGNOSTIC	ETAT	CRITERES
1° Traumatisme..... crânien simple	EV	.notion de traumatisme
	SU	.hémorragie auriculaire
	SU	.hémorragie nasale
	SU	.plaie du cuir chevelu
	SU	.hématome
2° Traumatisme..... crânien avec hématome	EV	.notion de traumatisme
	SU	.intervalle lucide entre l'accident et la perte de connaissance
	SU	.signes de latéralisation,
	SU	.signes d'hypertension intracrânienne
	PR	.Scanner: image d'hématome intracrânien
3° Coma éthylique.....	EV	.patient alcoolique notoire
	EV	.stigmates d'alcoolisme (angiomes stellaires, hépatomégalie, biologie)
	SU	.hypothermie

SU .hypotension
 SU .notion d'excès d'alcool
 SU .haleine: odeur d'alcool
 PR .alcoolémie élevée

4° Intoxication.....
 médicamenteuse

EV .notion de prise d'un médicament
 potentiellement dangereux, à savoir:
 .benzodiazépines
 .barbituriques
 .hypnotiques non barbituriques
 .tricycliques
 EV .tendance suicidaire
 EV .tremblements
 EV .myosis ou mydriase, réflexe photomoteur
 normal
 EV .hypothermie
 EV .notion de prise excessive d'un médicament
 potentiellement dangereux
 SU .coma calme, apyrétique, flasque, sans signe
 de localisation neurologique, réflexes O.T.
 faibles ou abolis
 PR .médicament dans le sang

5° Intoxication.....
 par drogue

EV .notion de patient drogué
 EV .traces de pique à l'avant-bras
 SU .notion de prise de drogue, myosis
 bilatéral
 PR .drogue dans le sang

6° Affection.....

neurologique (accident vasculaire cérébral (a.v.c.), tumeur, lésion cérébrale extensive)

- EV .notion d'a.v.c., de tumeur cérébrale ou de lésion cérébrale extensive dans les antécédents
- SU .signes de latéralisation (*), signes de localisation neurologique
- SU .EEG (signe d'atteinte neurologique),
- PR .Scanner (signe d'atteinte neurologique)

7° Hypoglycémie.....

du diabétique

- EV .patient diabétique traité par insuline ou sulfamidés hypoglycémisants, coma d'apparition brutale (<1h).
- SU .patient diabétique traité par insuline ou sulfamidés hypoglycémisants, coma d'apparition brutale (<1h), transpiration
- PR .patient diabétique traité par insuline ou sulfamidés hypoglycémisants, coma d'apparition brutale (<1h), hypoglycémie (*)
- EX .coma d'apparition lente (>1j)
- EX .glycémie normale ou hyperglycémie (*)

8° Acido-cétose.....

- EV .patient diabétique de type I, coma d'apparition lente (>1j), enfant
- SU .patient diabétique de type I, coma d'apparition lente (>1j), enfant, dyspnée de

Kussmaul (*)

- SU .patient diabétique de type I, coma
d'apparition lente (>1j), enfant, facteur
d'acidocétose (infection, stress, diminution
ou arrêt d'insuline ou de médicament
hypoglycémiant)
- SU .patient diabétique de type I, coma
d'apparition lente (>1j), enfant, haleine:
odeur d'acétone
- SU .patient diabétique de type I, coma
d'apparition lente (>1j), enfant, signes de
cétose (nausées, vomissements)
- SU .patient diabétique de type I, coma
d'apparition lente (>1j), enfant,
déshydratation (*)
- PR .coma d'apparition lente (>1j),
hyperglycémie (*), corps cétoniques
augmentés, glycosurie, acétonurie
- EX .glycémie normale ou hypoglycémie
- EX .coma d'apparition rapide (<1h)

9° Coma.....
hyperosmolaire

- EV .patient adulte, diabétique de type II, coma
d'apparition lente (>1j)
- SU .patient adulte, diabétique de type II, coma
d'apparition lente (>1j), déshydratation (*)
- SU .patient adulte, diabétique de type II, coma
d'apparition lente (>1j), signes
neurologiques (hémiparésie, signe de
Babinski)
- PR .coma d'apparition lente (>1j),
déshydratation (*), hyperglycémie sévère
- EX .coma d'apparition rapide (<1h)
- EX .glycémie normale

10° Désordres.....

hydro-électr.

et acido-bas.

PR .augmentation ou diminution du taux de Na+,
K+, Ca++, magnésium; hyper- ou hypo
-osmolalité; acidose (métabolique,
respiratoire), alcalose (métabolique,
respiratoire)

PR .gazs artériels: valeurs pathologiques

11° Insuffisance.....

rénale

SU .patient insuffisant rénal connu

SU .convulsions

PR .urée, créatinine augmentés

EX .urée, créatinine normales

12° Insuffisance.....

hépatique

SU .patient insuffisant hépatique connu

PR .enzymes hépatiques perturbés

EX .enzymes hépatiques normaux

13° Carence en.....

Vit B1

SU .patient alcoolique

PR .rétablissement de l'état de conscience
après administration de Vit B1 IV

14° Méningite.....

EV .notion de céphalées, vomissements

EV .signes d'infection locale ou générale
(examen clinique et syndrome inflammatoire à
la biologie)

SU .fièvre

SU .convulsions

PR .Ponction lombaire (P.L.) positive

15° Encéphalite.....

virale

- EV .infection virale les jours précédents
- EV .fièvre
- EV .hémiplégie
- EV .paraplégie
- EV .paralysie de nerfs crâniens
- PR .EEG, P.L.

16° Abscès cérébral.....

brat

- EV .notion de traumatisme des sinus
- EV .notion d'infection O.R.L. ou pulmonaire
- EV .notion de céphalées
- EV .notion de crise d'épilepsie
- EV .fièvre
- EV .biologie compatible avec une infection bactérienne
- SU .signes neurologiques, signes d'hypertension intracrânienne (*)
- SU .EEG compatible
- PR .Scanner

17° Septicémies.....

graves

- SU .fièvre, frissons
- PR .hémoculture positive
- PR .goutte épaisse (malaria)

18° Choc.....

- SU .pouls filant, tachycardie, hypotension, cyanose
- EX .T.A. et pouls normal

19° Hyperthermie.....

(coup de
chaleur)

EV .exposition au soleil
SU .température très élevée, peau sèche
EX .température normale ou diminuée

20°Etat post-.....
critique
(épilepsie)

EV .patient épileptique
SU .notion de crise d'épilepsie
PR .EEG
EX .coma de durée > 30'

5.7)GLOSSAIRE

.Déshydratation: sécheresse de la langue, peau sèche, pli cutané persistant, parfois [hypotension, pouls filant et tachycardie].
.Dyspnée de Kussmaul: respiration ample et rapide (due à l'acidose).
.Engagement: respiration de Cheyne-Stokes (alternance régulière de phases d'hyperpnée et de phases d'apnée plus brèves), anisocorie (diamètre pupillaire différent selon le côté), diminution du réflexe photomoteur, hypertonies.
.Hypertension intracrânienne: céphalées, bradycardie, hypertension artérielle, stase papillaire au fond d'oeil, signes d'engagement (*).
.Glycémie: mise en évidence par tigette (Dextrostix) ou prise de sang.
.Latéralisation: motricité (hémiplégie), tonus (hypertonies) motilité oculaire, diamètre pupillaire, réflexe photomoteur.

6) DEVELOPPEMENT

6.1) Introduction

Dans ce chapitre seront détaillées les différentes étapes de développement du prototype de système expert. Systématiquement, nous décrirons l'objectif de l'étape, les informations en entrée, les produits en sortie, le type d'activité, la méthodologie et les outils employés ainsi que les difficultés rencontrées. Dans la mesure du possible, les décisions et les choix seront explicités et justifiés. L'accent sera mis sur l'explication du raisonnement suivi plus que sur la simple description des étapes. Si des hypothèses ont été faites, elles seront précisées. Les différents stades de développement définis sont l'analyse d'opportunité, la spécification fonctionnelle, la conception et le codage.

6.2) L'analyse d'opportunité

Objectif de l'étape

Cette première phase a pour but d'étudier le cadre général du projet en définissant ses limites et en précisant les objectifs poursuivis.

L'environnement du projet et ses objectifs généraux ayant déjà été définis au chapitre 4, nous nous attacherons essentiellement à l'étude des besoins en information et des conditions à satisfaire par le futur système.

Information en entrée

L'information en entrée est la situation existante. Quand le service des urgences reçoit un malade comateux, des décisions importantes sont

prises et des gestes vitaux sont posés. L'information qui s'y rapporte est confinée dans le dossier médical du malade ou est fournie par le personnel qui a vécu le cas "en direct".

Produits en sortie

L'étude de la situation existante permet de dégager les points critiques quant aux besoins en information.

- 1° Il faut tout d'abord insister sur le caractère urgent de la démarche. La prise de décision est conditionnée par le degré de priorité que peut avoir une information sur une autre. Il est par exemple fondamental de vérifier l'état des fonctions vitales avant d'envisager de rechercher le(s) diagnostic(s). Un algorithme de travail serait souhaitable mais n'est pas réaliste vu le grand nombre d'alternatives possibles. Il faudrait cependant pouvoir tenir compte d'une séquentialité des gestes à poser en fonction de leur priorité (degré d'urgence). Un autre facteur de séquentialité à considérer est le scénario interrogatoire/examen clinique/examens complémentaires commun à toute consultation médicale. C'est ainsi qu'il est impensable d'envoyer le patient au scanner sans l'avoir préalablement examiné. La mise au point d'un patient comateux est une démarche pas à pas où l'ordre des étapes, même si il n'est pas figé, doit être respecté.
- 2° D'autre part, la mise au point d'un patient comateux implique un raisonnement à partir de données incomplètes. Il faut pouvoir prendre des décisions sur base d'hypothèses diagnostiques ou de symptômes partiels. La faculté d'évoquer des hypothèses est donc indispensable car le plus souvent, on ne peut se permettre d'attendre des confirmations formelles avant d'agir. Or, il est très difficile d'avoir à l'esprit toutes les causes possibles de coma et leurs symptômes d'appel. Il serait donc très utile d'avoir un aide-mémoire qui puisse le cas échéant faire penser à une hypothèse diagnostique à partir de peu d'informations.
- 3° Un troisième point très important à souligner est le type d'information utile dans la mise au point. Dans un premier temps, il s'agit de déterminer les gestes les plus urgents par ordre de

priorité ou selon le bon sens pratique. Ceux-ci sont peu nombreux et, pour la plupart, sont posés quelque soit la cause de coma. Une autre fonction importante est de donner, à partir d'un ensemble de signes, l'état diagnostique du patient c'est à dire les causes de coma qui sont exclues, simplement évoquées, suspectées ou prouvées. On peut résumer la situation en affirmant que la détermination des premiers gestes relève de l'aide à la décision tandis que la deuxième fonction est de l'ordre de l'aide au diagnostic.

Type d'activité

L'analyse de l'existant a nécessité une révision des données médicales relatives à la mise au point d'un patient comateux en urgence et au diagnostic différentiel des causes de coma. Les résultats de cette recherche ont été exposés au chapitre précédent.

D'autre part, la connaissance empirique, basée sur l'expérience du médecin, a été obtenue par interviews et par étude des dossiers médicaux. La démarche en urgence a pu être reconstituée et les besoins en information identifiés.

Difficultés

La principale difficulté rencontrée dans cette étape a été de déterminer la nature des besoins en information. Une notion clé est de ne pas considérer la mise au point d'un patient comateux seulement comme une tâche de recherche de diagnostic. Il est tout aussi important de considérer les gestes urgents à poser. Les deux fonctions mises en évidence (aide au diagnostic, aide à la décision) sont de nature très différentes (la première est déclarative, la seconde plus procédurale) mais sont indissolubles dans le temps.

Une autre difficulté, en partie liée à la première, est de tenir compte du temps, de la séquentialité. Il n'est pas aisé de formaliser la

dynamique des événements car il n'existe pas de scénario figé. Deux facteurs de séquentialité ont pu malgré tout être identifiés: la priorité selon le degré d'urgence et le bon sens pratique (examen clinique avant les examens complémentaires).

6.3) SPECIFICATION FONCTIONNELLE

Objectif de l'étape

Cette deuxième étape consiste à spécifier les fonctions que devront accomplir le système expert. Elle se base sur les résultats de l'étape précédente dans le sens de répondre à des besoins en information. De plus, il convient de définir avec précision ce dont sera capable le système expert mais aussi en indiquer ses limites.

Information en entrée

Les résultats de l'analyse d'opportunité. Retenons essentiellement les conditions à satisfaire:

- 1° Tenir compte de la séquentialité de la mise au point d'un patient comateux.
- 2° Le système devra être capable de tirer des conclusions à partir de données incomplètes.
- 3° Les résultats à fournir sont de deux nature: propositions de mesure (aide à la décision), état diagnostique du patient (aide au diagnostic).

Produits en sortie

Les fonctionnalités du système expert devront répondre aux conditions ci-dessus. L'énumération des grandes fonctions peut s'établir comme suit:

- 1° Représenter la connaissance scientifique et la connaissance empirique de manière explicite.
- 2° Modéliser une démarche procédurale propre à la mise au point d'un patient comateux.
- 3° Modéliser une fonction diagnostique sur base d'information incomplète.
- 4° Offrir une interface à l'utilisateur sous forme d'un dialogue proche du langage naturel.
- 5° Permettre à l'utilisateur d'accéder aux connaissances et aux critères diagnostiques.
- 6° Pouvoir donner des justifications aux résultats fournis.

La définition de ces grandes fonctionnalités appelle plusieurs commentaires.

D'abord, il faut remarquer que représenter la connaissance de manière explicite est indispensable pour rencontrer les fonctions 5° et 6°.

D'autre part, il faut rappeler le choix de séparer les deux fonctionnalités 2° et 3° qui sont de nature différente même si elles sont liées en pratique. En effet, modéliser une fonction diagnostique n'est pas spécifique de l'application (mise au point d'un patient comateux) mais commun à toute consultation médicale. Par contre, la démarche procédurale est propre au sujet qui nous occupe.

Par ailleurs, il n'est fait aucune mention de contraintes de performances pour la simple raison que l'hypothèse est faite de les ignorer.

Enfin, le lecteur aura remarqué que les fonctionnalités ressemblent fort aux potentialités d'un système expert. Ce serait trahir la réalité que de nier que, dès la début, la conception d'un système expert était à l'esprit. Cette étape de spécification fonctionnelle a été consacrée en partie à la définition d'un commun dénominateur entre les besoins en informations et les possibilités offertes par un système expert. Ceci est en fait en corrélation avec l'objectif de produire un système à caractère

expérimental.

Notons pour terminer que l'accent est mis sur les premiers gestes en urgence et la recherche diagnostique, le traitement causal (après l'établissement d'un diagnostic) et le pronostic ne faisant pas partie du cadre de ce travail.

Type d'activité

La principale activité développée dans cette étape a été une redéfinition des résultats de l'analyse d'opportunité en termes de spécifications, c'est-à-dire en précisant ce qu'il y avait lieu de faire.

Méthodologie

Nous n'avons pas pu nous baser sur les modèles largement répandus tels le schéma entité/association ou le modèle de la dynamique. En effet, nous pensons que l'application qui nous occupe est de nature trop informelle, trop peu structurée pour en profiter.

6.4) CONCEPTION

Objectif de l'étape

Obtenir une conceptualisation (définition des concepts, des objets, des relations) et une formalisation (mode de représentation des concepts). Ces deux premières phases définissent le cadre de travail pour la réalisation des spécifications.

Information en entrée

Les données de l'application (exposées au chapitre 5) et les six fonctionnalités définies à l'étape précédente constituent les informations de départ.

Produits en sortie

1) Phase de conceptualisation

Concepts, valeurs, relations

Les concepts à manipuler sont les suivants:

- .donnée d'ordre général (âge, sexe,...)
- .antécédent (passé du patient)
- .symptôme (plaintes, donnée de l'épisode actuel)
- .signe clinique (donnée de l'examen clinique)

- .examen complémentaire (radiographie, p.ex.)
- .signe (donnée pouvant être un des cinq concepts précités)
- .syndrome (ensemble de signes)
- .diagnostic
- .traitement

Ces concepts fondamentaux peuvent prendre certaines valeurs.

Concept	Valeurs
signe	vrai (présent) faux (absent)
donnée quantifiable	normal diminué augmenté

diagnostic

exclu

évoqué

suspecté

prouvé

traitement

indiqué

contre-indiqué

Il est important de noter que ces concepts ont été obtenus après une étude approfondie des connaissances et des raisonnements médicaux en jeu.

C'est ainsi que le raisonnement qualitatif, très utilisé en médecine, sera préféré au raisonnement quantitatif chaque fois que c'est possible. Un résultat de dosage sanguin, représenté par une donnée chiffrée est le plus souvent interprétée par le médecin en zones de décision plus ou moins précises (augmenté/ diminué/normal p.ex.). Il faut de plus admettre que la prise en compte d'une conversion d'une donnée numérique en appréciation qualitative est souvent malaisée. C'est pourquoi, nous nous contenterons le plus souvent de considérer le raisonnement qualitatif.

D'autre part, ce mode de définition des concepts permet de préparer le terrain pour modéliser le raisonnement à partir de données incomplètes. En effet, la valeur prise par le concept diagnostic pourra être définie même en l'absence de certaines données grâce à l'échelle progressive des valeurs de vérité. Certains signes pourront donc évoquer ou faire suspecter un diagnostic sans pouvoir l'exclure ou le prouver.

Les relations entre ces concepts peuvent être définies sans autre limitation que notre imagination. Citons en les principales:

- 1°critère de diagnostic: tel signe (ou groupe de signes) évoque, fait suspecter, prouve ou exclut tel diagnostic.
- 2°définition de syndrômes: tel groupe de signes définit un syndrome.
- 3°proposition de mesures: tel signe (ou groupe de signes) nécessite tel examen complémentaire.

4°traitement: tel diagnostic prouvé impose tel traitement;
ou: la présence de tel signe contre-indique tel
traitement.

2)Phase de formalisation

Cette phase a pour but de préciser un mode de représentation des concepts définis dans la phase de conceptualisation.

Il faut mentionner d'emblée que le logiciel Turbo Prolog a été dès le départ adopté. Il est donc évident que les choix de formalisation ont été influencés par les possibilités offertes par ce langage (voir chapitre 7:le langage Prolog).

Les concepts définis plus haut seront représentés par une variante du classique triplet <objet,attribut,valeur>:

<concept,nom,valeur de vérité>

on aura, par exemple: <diagnostic,coma diabétique,suspecté>

ou encore: <examen complémentaire,glycémie,normale>.

Pour représenter les relations entre ces concepts, nous avons opté pour les règles de production du type:

Si liste-et de conditions

Alors conclusion

Le format pour celles-ci est défini comme suit (selon les conventions BNF):

```
règle ::= numéro de règle, liste-et de conditions, conclusion
numéro de règle ::= entier
liste-et de conditions ::= condition ! condition, liste-et
de conditions
condition ::= triplet
conclusion ::= triplet
triplet ::= concept, nom, valeur de vérité
```

Ce mode de représentation est explicite et répond donc à la première fonctionnalité définie à l'étape précédente. D'autre part, pour réaliser les autres spécifications, il était indispensable de définir les concepts de base et de les représenter, ce qui a été fait dans les deux premières phases (conceptualisation et formalisation).

3) Phase de réalisation des spécifications

3.1) Démarche procédurale

La mise au point d'un patient comateux en urgence nécessite, comme on l'a vu au chapitre 5 (voir 5.3: les premiers gestes), une démarche procédurale. Celle-ci est sensiblement différente de la fonction diagnostique, s'attachant à la proposition de gestes à poser de manière suffisamment stéréotypée pour pouvoir être représentée par un algorithme décisionnel. Celui-ci n'est en fait qu'une succession de questions/réponses avec peu d'alternatives. Il s'obtient de manière évidente à partir de la description des premiers gestes faite au chapitre 5.

Cet algorithme est à la fois une aide à la décision et un questionnaire

qui permet de recueillir une foule d'informations. Celles-ci seront utilisées par la fonction diagnostique développée dans la partie qui suit.

3.2) Fonction diagnostique

Modéliser une fonction diagnostique sur base d'information incomplète, telle est la spécification à réaliser.

Dans un système expert utilisant les règles de production, deux méthodes de résolution existe: le chaînage avant et le chaînage arrière. La première part des données disponibles (les signes) et procède à des déductions en utilisant les règles applicables jusqu'à obtention d'une conclusion (diagnostic). L'avantage est de tenir compte d'emblée des données mais les règles sont activées à l'aveugle, sans garantie de déduire des diagnostics.

A l'inverse, la méthode de chaînage arrière part des buts (diagnostics) et consiste à essayer de prouver les conditions qui y mènent. L'avantage est la prise en compte d'emblée des règles de diagnostics, le désavantage étant de ne pas tenir compte des informations disponibles pour orienter la recherche et de procéder de manière exhaustive.

Si l'on observe la manière de procéder dans une démarche diagnostique (voir 2.1.1: la démarche du médecin), on remarque que bien souvent, il est utile d'évoquer d'abord des hypothèses diagnostiques à partir des premiers signes (en chaînage avant) et d'essayer de les confirmer ensuite par l'évaluation des critères de diagnostic (par chaînage arrière).

C'est la raison pour laquelle nous avons opté pour une recherche basée sur les deux types de chaînage.

Chaînage avant (figure 7)

Après la démarche procédurale (gestes en urgences), la base de faits contient des informations sur l'état du patient (symptômes, signes cliniques,...) mais sans notion de diagnostic. Le module de chaînage avant va, à partir de ces informations, utiliser les règles de la base de connaissances pour déduire tout ce qui peut l'être.

Des règles de diagnostics pourront, le cas échéant, être activées et faire passer un diagnostic dans l'un des états suivants: exclu, évoqué, suspecté, prouvé, complet. Ces quatre derniers états sont ordonnés (évoqué < suspecté < prouvé < complet) de telle manière que seul l'état le plus fort sera retenu. Les hypothèses diagnostiques sont les diagnostics dont l'état est évoqué ou suspecté.

Comme le montre la figure ci-dessous, le module de chaînage avant va enrichir la base de faits de toutes les conclusions qu'il aura pu déduire des informations dont il dispose. On parle de chaînage avant parce que l'on part des prémisses des règles pour arriver aux conclusions.

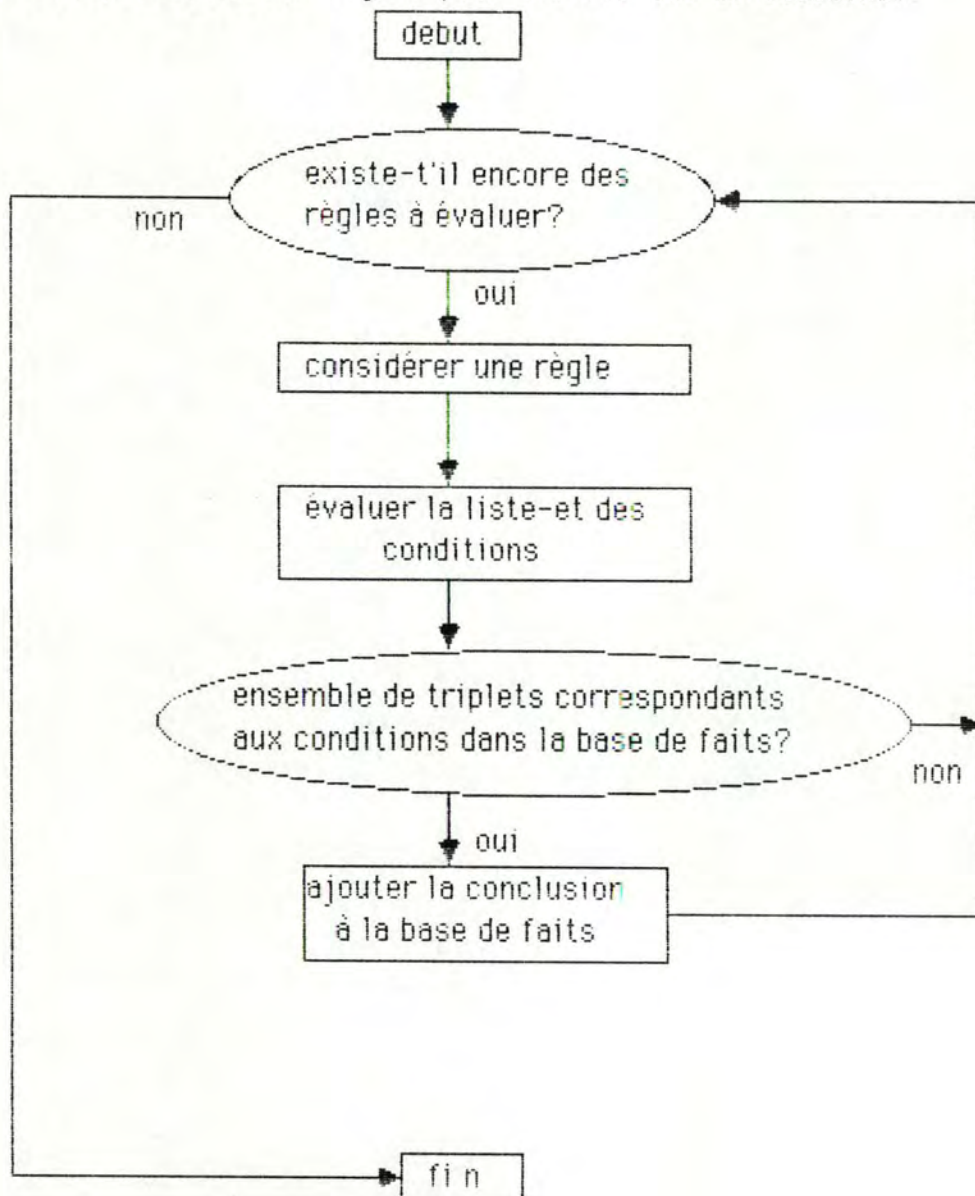


figure 7: Chaînage avant

Chainage arriere (figure 8)

Ce module, considérant en entrée les hypothèses diagnostiques (diagnostics dans l'état évoqué ou suspecté) de la base de faits va utiliser les règles adéquates pour tenter de faire passer ces hypothèses dans l'état exclu ou dans un état supérieur. On parle de chainage arriere car on part de la conclusion de la règle avant de considérer les conditions. Celles-ci sont évaluées une à une de la manière suivante: voir si le triplet correspondant à la condition n'est pas déjà dans la base de faits; sinon tenter de le déduire; en cas d'échec, poser la question à l'utilisateur.

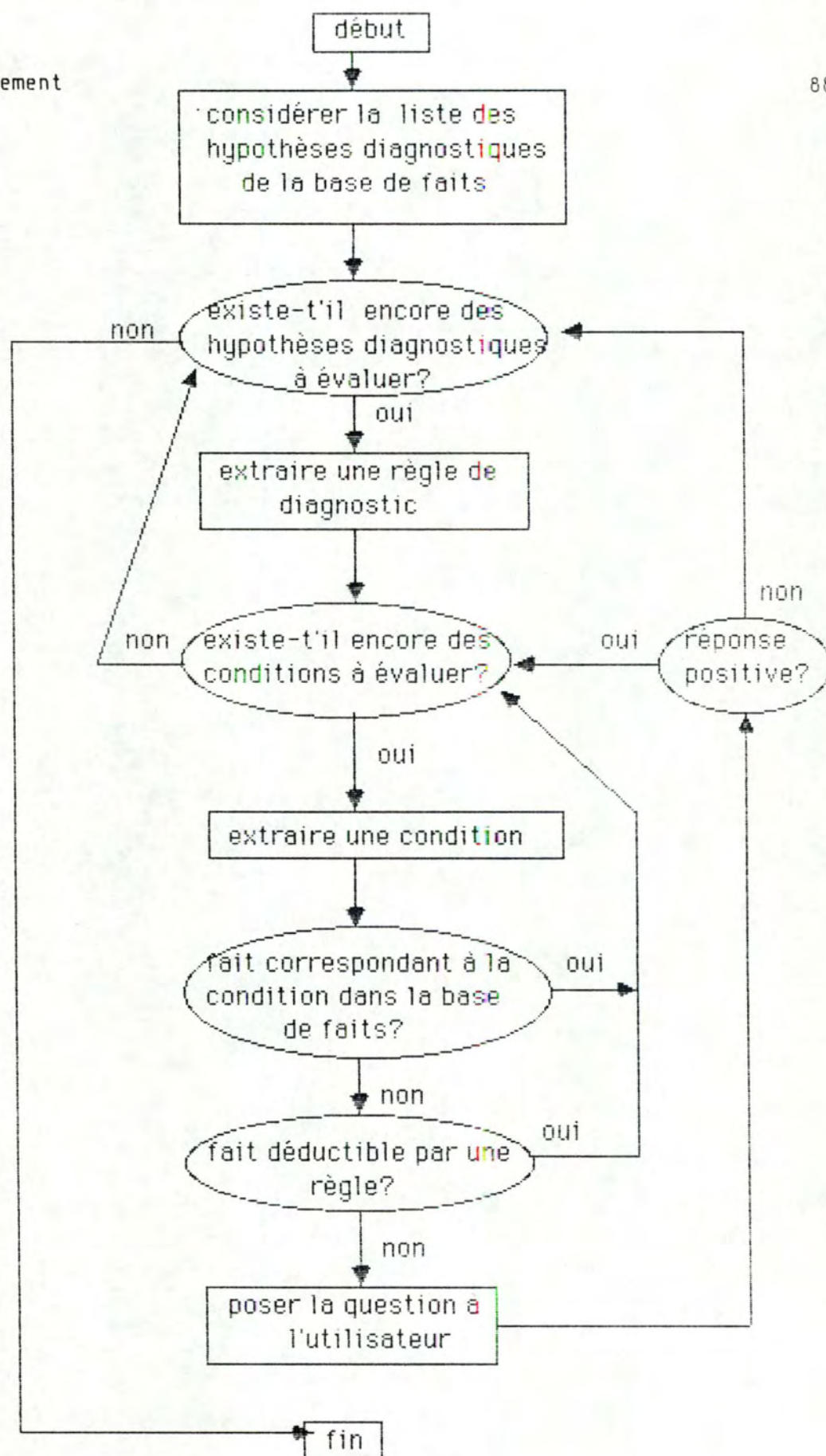


figure 8: Chainage arrière

3.3)Interface

Toute information étant représentée sous forme d'un triplet <concept,nom,valeur>, un seul module permet d'écrire un résultat quelqu'en soit sa nature.

Pour poser une question à propos d'un fait, il convient d'abord de rechercher dans la base de connaissances le type de domaine auquel il appartient. Nous appelons domaine l'ensemble des valeurs que peut prendre un concept. C'est ainsi que l'on définit essentiellement:

1°domaine simple: question simple du type oui/non

p.ex.: triplet(antécédent,diabète,vrai)

domaine simple (oui/non)

notion de diabète dans les antécédents?

2°domaine "menu": question du type "menu"

2.1°menus standards

p.ex.: triplet(examen complémentaire,glycémie,diminuée)

domaine menu standard "quantifiable"

glycémie: normale?

diminuée?

augmentée?

2.2°menus dont les valeurs sont à énumérer

p.ex.: triplet(examen clinique,fond d'oeil,stase papillaire)

domaine menu "à énumérer"

fond d'oeil: normal?

rétinopathie diabétique?

stase papillaire?

3.4)Accès à la base de connaissances

Les règles sont écrites de manière unique et les faits élémentaires ayant tous la même structure de triplet, il est aisé de rendre la base de connaissances accessibles à l'utilisateur. En effet, elle peut être considérée comme une base de données relationnelle où la table "règle" comprend les colonnes "numéro"(identifiant), "conditions" et

"conclusion".

En pratique, il est utile de prévoir des requêtes toutes faites qui répondent aux principaux besoins en information tels la liste des diagnostics, l'ensemble des signes d'un diagnostic, les critères pour prouver ou exclure un diagnostic.

3.4) Module explicatif

Une des caractéristiques des systèmes experts est de pouvoir expliquer les résultats fournis et de justifier les questions posées. Une des informations, contenue dans chaque règle est un numéro qui l'identifie. Par ce moyen, il est possible de mémoriser la règle qu'on est en train de prouver et, via l'interface, donner la possibilité à l'utilisateur de répondre "pourquoi?" à la question qu'on lui pose et de cette façon lui afficher (dans un langage naturel) cette règle. De plus, l'utilisateur pourra inspecter la règle qui a permis de prouver un résultat en demandant "comment?".

Difficultés

La principale difficulté a été de modéliser la fonction diagnostique. L'optique que nous avons choisie (évocation d'hypothèses par chaînage avant, évaluation par chaînage arrière) correspond schématiquement à la réalité mais possède quelques lacunes en raison du scénario figé que nous nous sommes fixés. Parfois, il est intéressant d'évaluer plusieurs hypothèses parallèlement, ce qui n'est pas possible dans notre système. Par ailleurs, ne prendre en considération que les hypothèses évoquées ou suspectées risquent de négliger une hypothèse plus ou moins cachée a priori. Le problème est de trouver une heuristique qui puisse diminuer l'arbre de recherche sans entacher la fonction diagnostique d'approximations trop larges. La solution évoquée ici n'est qu'une approche et ce problème mériterait une étude plus approfondie.

6.5) Codage (voir programme en annexe)

Information en entrée

Les résultats de la conception constituent les informations en entrée à savoir le mode de représentation des règles et des faits élémentaires, la démarche procédurale, la fonction diagnostique (chainage avant, chainage arrière), l'interface et le module explicatif.

Produits en sortie

En sortie, il faut disposer d'une solution exécutable sur un micro-ordinateur Olivetti M24 de 640K de mémoire centrale.

Outils

Le logiciel utilisé est le Turbo Prolog (version 1.1).

Difficultés

Le codage s'est fait de manière modulaire et l'intégration des différents modules a posé des problèmes de performances. Il est apparu à l'exécution des dépassements de capacités de pile et de taille mémoire. Nous n'avons pas eu la possibilité de procéder à une étude de performances mais il semble que deux facteurs entrent principalement en ligne de compte: la mémorisation des points de backtracking et le gestionnaire d'écran (fenêtres, menus).

7) LE LANGAGE PROLOG

7.1) LA PROGRAMMATION LOGIQUE

L'idée centrale de la programmation logique peut être comprise à partir de la comparaison avec la programmation classique. Dans cette dernière, on peut être amené à spécifier un traitement en décrivant les propriétés des données et celles des résultats (spécification par pré- et post-assertions). En généralisant ce concept, on peut considérer une relation entre les données et les résultats. En généralisant encore, les notions de données et résultats disparaissent pour être remplacées par le terme d'objets reliés par une relation.

Comme on le voit, la programmation logique ne considère pas un programme comme une séquence d'actions qui à partir de données vont produire les résultats attendus. Le point de vue est sensiblement différent: le "programme" est vu comme un ensemble de relations entre objets qui décrivent des propriétés d'une situation. Par nature, l'approche classique, dite algorithmique, s'attache à la manière de résoudre un problème par un enchaînement d'actions élémentaires tandis que la programmation logique cache cet aspect procédural en mettant l'accent sur l'aspect déclaratif.

Sans rentrer dans les détails, notons qu'il existe en programmation logique trois notions de base: le fait, la règle et la question. La structure de donnée fondamentale est le terme qui est une constante, une variable ou un terme composé. Ce dernier est constitué d'un foncteur et d'un certain nombre d'arguments qui sont eux mêmes des termes. Un terme composé a donc la forme suivante:

$$f(a_1, a_2, \dots, a_n) \quad \text{où } f \text{ est un foncteur}$$

et a_i ($1 \leq i \leq n$) un argument qui

est aussi un terme

Les faits expriment une relation entre objets, par exemple:

père-de(pierre,paul)

La règle est une formulation du type:

A si B1,B2,...,Bn où A est un terme composé
appelé tête de règle et les Bi sont des termes composés qui forment
le corps de la règle
Il est intéressant de noter qu'un fait n'est rien d'autre qu'une règle
sans corps.

La question, elle, est une règle sans tête et porte sur l'existence
d'une relation définie entre les objets donnés. Le but est de déterminer
si ce qui est exprimé dans la question est oui ou non une conséquence
logique du programme. Si ce n'est pas le cas, cela ne signifie pas que ce
qui est exprimé dans la question est faux mais que la relation entre
objets décrite n' a pu être prouvée à partir des faits et des règles.

Schématiquement, un programme logique est constitué d'un ensemble de
règles et de faits qui décrivent des relations entre objets. On soumet à
ce programme une question et un mécanisme indépendant va se charger de
sélectionner et de traiter les faits et les règles dans le but de
déterminer si la relation proposée est dérivable ou non. Volontairement,
le terme de "mécanisme", qui est peu explicite, a été employé pour ne pas
décrire en profondeur la manière dont les choses se déroulent.
L'important à ce stade est de montrer qu'en programmation logique,
l'aspect procédural est caché de manière à fournir au programmeur la
possibilité de voir le problème de manière descriptive (approche
déclarative).

7.2)INTRODUCTION AU LANGAGE PROLOG

Le terme "Prolog" vient de l'anglais Programming in logic. Il s'inspire

donc des principes de la programmation logique. Il a été développé pour la première fois en 1972 par A. Colmerauer à l'Université de Marseilles. Le premier compilateur Prolog a été lui réalisé à l'Université d'Edinburgh. Actuellement, il en existe toute une série de versions tournant sur divers types d'ordinateurs. Récemment, plusieurs versions peu coûteuses pour micro-ordinateurs sont apparues, telles que micro-Prolog et, plus récemment Turbo-Prolog.

Le langage Prolog se base sur la logique des prédicats du premier ordre (voir partie I: 3.2.3: techniques de représentation). Les clauses (règles, faits et questions) sont des formules du premier ordre avec au plus un littéral non nié; on parle de clauses de Horn. Pour tirer des conclusions, ces clauses sont soumises au principe de résolution (dériver de deux clauses une troisième appelée résolvante) restreint à ce que l'on appelle la "L.U.S.H. resolution" (Linear resolution with Unrestricted Selection Function for Horn Clauses). Celle-ci est une résolution linéaire, c'est-à-dire où la résolvante est obtenue à partir d'une clause et de la résolvante précédente, ce qui correspond à une recherche en profondeur d'abord. D'autre part, le choix des clauses se fait selon l'ordre d'apparition dans le texte et le littéral complémentaire candidat à l'effacement est celui le plus à gauche.

Ces considérations permettront de mieux comprendre le fonctionnement de l'interpréteur Prolog que nous étudierons dans les grandes lignes dans la partie 8.4. Basé sur le principe de résolution, il fonctionne de manière stéréotypée (cfr LUSH-resolution) comme un démonstrateur de théorèmes en travaillant selon le principe de preuve par réfutation.

7.3) SYNTAXE DU LANGAGE PROLOG

Nous choisissons ici les conventions d'écriture du Prolog d'Edinburgh dont est issu le logiciel Turbo Prolog utilisé dans la réalisation de ce mémoire.

Les notions de base sont les suivantes:

1°Termes: .constante (caractère, entier,...)

.variable (représentée par un mot commençant par une majuscule: X,Personne p.ex.)

.terme composé: $f(a_1, a_2, \dots, a_n)$

où f est un foncteur

et a_i ($1 \leq i \leq n$) un argument qui est aussi un terme

.liste: [P|R]

où chaque élément est un terme, P étant le premier et R la liste des éléments restants

2°Clauses: .règle: $C :- A_1, A_2, \dots, A_n.$

où C et les A_i sont des termes composés

.fait: $C.$ (règle sans antécédent) (C est un terme composé)

.question (ou but): $Q_1, Q_2, \dots, Q_n.$ (règle dont le conséquent est vide)

(les Q_i sont des termes composés)

3°Procédure: ensemble de règles ayant même foncteur

4°Programme: ensemble de procédures

Il est intéressant de noter que les termes composés peuvent à eux seuls représenter

.un fait: $\langle(3,8)\rangle$ p.ex.

.une structure de donnée telle que:
 personne(nom,prénom,adresse)

.un arbre syntaxique:

 if cond $(X>0)$ then $Y:=Y+X$

 pourrait être exprimé sous la forme suivante:

if(cond($>(X,0)$),affect(var(Y),expr($Y+X$))).

Il est enfin possible de combiner des termes composés pour exprimer un fait à propos d'un objet lui même composé, comme par exemple:

père-de (personne(dupont,jean,av. de la gare),
 personne(dupont,jacques,rue longue)).

Une attention toute particulière doit être prêtée à la notion de variable qui diffère de celle utilisée en programmation classique. En effet, une variable ne doit pas être vue comme une "case mémoire" à laquelle on peut affecter une valeur qu'elle garde tant qu'une autre valeur ne lui a été affectée (assignation dite destructive). En Prolog, la portée d'une variable se limite à l'exécution d'une règle et la liaison avec une valeur est de nature différente, on parle d'"instantiation" et il n'existe pas d'affectation destructive. On comprendra mieux le rôle des variables en Prolog en connaissant la manière dont elles sont traitées par l'interpréteur, ce qui fait l'objet de la partie qui suit.

7.4) MECANISMES DE PROLOG

Le "mécanisme indépendant" (mentionné plus haut) est capable de manipuler les clauses pour essayer de prouver le but (question) qui lui est soumis et est appelé interpréteur Prolog ou moteur d'inférence. C'est lui qui choisit les règles applicables, organise les déductions qui en découlent, revient en arrière si des voies de recherche s'avèrent infructueuses. Prolog offre donc non seulement un mode de représentation de connaissances (les faits et les règles) mais aussi un mécanisme de raisonnement (l'interpréteur). Le travail de ce dernier est en principe caché au programmeur en vertu de l'approche déclarative de la programmation logique. Cependant, quand on programme en Prolog, il faut tenir compte de la manière de travailler de l'interpréteur et c'est la raison pour laquelle il est important de comprendre les mécanismes qui le sous-tendent à savoir le chaînage arrière, l'unification, le backtracking et ce que l'on appelle les primitives extra-logiques (le "cut" et le "fail").

7.4.1) CHAINAGE ARRIERE

Les règles Prolog s'écrivent sous la forme:

$$B \text{ :- } A1, A2, \dots, A_n$$

qui peut être comprise de la manière suivante: si les termes $A1, A2, \dots, A_n$ sont vrais, alors on peut en déduire que B l'est aussi. Un très grand nombre de connaissances peuvent être représentées de cette façon. Soit la règle isolée:

$$\text{grand-parent}(X, Z) \text{ :- } \text{parent}(X, Y), \text{parent}(Y, Z)$$

Dans cet exemple, pour prouver que $\text{grand-parent}(X, Z)$ est vrai, il suffit de démontrer que $\text{parent}(X, Y)$ et $\text{parent}(Y, Z)$ sont vrais tous les deux. La dynamique du raisonnement est donc d'essayer de prouver un but

par réduction de ce but en sous buts. Cette méthode est appelée "raisonnement régressif" et s'apparente à la technique qui porte le nom de "chaînage arrière". Lorsque l'utilisateur (ou le programme lui même) donne le but à satisfaire, l'interpréteur déclenche une chaîne d'inférences en décomposant s'il y a lieu, le but en sous-buts. Le premier de ceux-ci sera à son tour considéré comme un but et décomposé à son tour. Ce mode d'action s'apparente à la technique de recherche en profondeur d'abord. Illustrons ce concept à l'aide d'un petit exemple.

Supposons un programme Prolog qui comprend les clauses suivantes:

p :- q1, q2.

q1 :- r1, r2.

r1.

r2.

Si l'utilisateur pose la question "p", l'interpréteur va rechercher s'il existe une clause dont la tête correspond à la question. Le parcours des règles se faisant de haut en bas (et cela systématiquement de manière stéréotypée), le système tombe sur la première clause qui indique que p sera satisfait si les deux sous-buts q1 et q2 le sont aussi. Le parcours des sous-buts dans une même règle se faisant de gauche à droite, le moteur va donc tenter de satisfaire q1. Celui-ci le sera si r1 et r2 le sont et ainsi de suite. On voit facilement que q1 peut être satisfait. Il ne reste plus qu'à prouver q2, ce qui est immédiat dans cet exemple.

7.4.2) L'UNIFICATION

Unifier deux termes consiste à les mettre en correspondance pour vérifier s'ils sont identiques à un jeu de substitution de variables près. L'unification s'effectue de manière syntaxique en comparant les termes; cela signifie qu'elle réussit si les règles suivantes sont respectées:

.une constante ne peut s'unifier qu'à elle même

.une variable peut s'unifier à n'importe quel terme (si

l'unification se fait avec une constante et que la variable est libre, on dit qu'elle est instanciée (liée) à cette valeur).

.un terme composé s'unifie à un autre terme composé et cela

uniquement si les foncteurs sont identiques, le nombre d'arguments est le même et ces derniers s'unifient deux à deux.

Supposons qu'un but est satisfait si la liste des sous-buts b_1, b_2, \dots, b_n l'est également et qu'il existe une règle de la forme $c :- a_1, a_2, \dots, a_m$. Cette dernière sera appliquée s'il existe un jeu de substitution qui permettent l'unification de b_1 avec c . Si c'est le cas, la nouvelle liste de sous-buts devient:

$\langle a_1, a_2, \dots, a_n, b_2, \dots, b_n \rangle$.

L'unification est donc un mécanisme puissant qui permet de faire avancer les inférences en trouvant les règles applicables et résume à elle seule des notions telles que passage de paramètres, instantiation de variables et test d'égalité.

7.4.3) BACKTRACKING

Pour satisfaire un but, il faut essayer de satisfaire les différents sous-buts découverts. Cela revient en fait à définir un algorithme de recherche selon un arbre ET/OU. Pour répondre à un but (question), Prolog doit donner toutes les solutions possibles qui satisfont ce but. Lorsqu'une règle aura réussi en totalité, il faudra faire un rétroparcours (backtracking) et essayer d'autres instantiations de variables de façon à trouver les autres solutions éventuelles. De même, certains parcours de l'arbre peuvent conduire à un échec. Il sera donc ici nécessaire de remonter jusqu'au noeuds dont toutes les possibilités n'ont pas encore été explorées.

7.4.4) LES PRIMITIVES EXTRA-LOGIQUES

Les primitives extra-logiques sont au nombre de deux (le "cut" et le "fail") et sont appelées de la sorte parce qu'elles s'opposent au principe de la programmation logique. En effet, elles constituent des moyens donnés au programmeur pour influencer le comportement de l'interpréteur en opposition avec l'approche déclarative qui veut que l'on décrive les relations entre objets sans considération de la manière dont ils seront manipulés.

1. Le "cut"

Cette primitive permet de signaler à l'interpréteur Prolog qu'il est inutile de considérer certains choix lors du backtracking. Cela permet de limiter le nombre de solutions à un problème ou d'éviter à l'interpréteur de s'engager sur des voies dont on sait à l'avance qu'elles ne contribueront jamais à une solution. Ce mécanisme permet donc des gains de performances (temps, place) en épargnant la mémorisation de points de backtracking.

Le "cut" (dont la notation est "!") se présente comme un prédicat sans argument qui réussit immédiatement et ne pourra plus l'être une deuxième fois: passé un "cut", il n'est plus possible de "backtracker".

Illustrons le à l'aide d'un exemple: il est possible de construire un prédicat "if" qui fonctionne à la manière du classique "if...then...else...". Il peut s'écrire de la manière suivante:

```
if(C,T,E) :- C!,T.
```

```
if(C,T,E) :- E.
```

Cet exemple exprime que si la condition C est vraie, on cherche à prouver T et la deuxième règle ne sera pas prise en compte. Si C échoue, on ne traverse pas le "cut" et la règle suivante sera évaluée.

2. Le "fail"

Il s'agit d'un opérateur (noté "fail") sans argument qui, contrairement au "cut", échoue toujours. Il est utilisé pour forcer un backtracking. Cela peut parfois se révéler utile, notamment en le combinant avec le

"cut" pour forcer la terminaison d'une procédure:

ex.:

a :- b,!,fail.

a:- c.

Les deux clauses échoueront si b réussit.

7.5) SYSTEMES EXPERTS ET PROLOG

Prolog est considéré comme un langage lié à l'intelligence artificielle et en particulier utile à la réalisation d'un système expert (SE). Au delà des clichés, il est important de montrer en quoi ce langage présente des facilités pour réaliser certaines fonctions propres aux SE.

Tout SE comprend une base de connaissance et un moteur d'inférence. IL nécessite un traitement symbolique car les connaissances sont représentées au moyen de symboles. La connaissance étant granulaire, il faut un mécanisme de recherche pour accéder à ces multiples unités de connaissance. Enfin, un module explicatif permet de justifier les questions ou les conclusions à l'utilisateur.

Le plus souvent, les connaissances sont représentées à l'aide de règles de production et le mécanisme d'inférence le plus utilisé est le chaînage arrière qui considère une donnée comme un but à satisfaire. Ce dernier point est particulièrement vrai pour les SE de diagnostic, ce qui permet d'envisager les hypothèses diagnostiques les unes après les autres.

Essentiellement, il faut donc, pour réaliser un SE:

1° un mode de représentation des connaissances

2° un mécanisme d'inférence

3° un mécanisme de calcul dans un espace de recherche

Voyons quels sont les potentialités et les limites de Prolog pour ces trois points de vue.

Prolog, par sa syntaxe déclarative, facilite la représentation de la connaissance sous forme granulaire. Nous ne reviendrons plus sur ce point même s'il représente sans conteste un avantage considérable de Prolog par rapport aux langages procéduraux dans la réalisation d'un SE. Il faut

cependant reconnaître qu'il est difficile de construire en Prolog des objets complexes telles que des hiérarchies. Prolog est donc adéquat pour la représentation de règles de production mais convient moins si on utilise des objets structurés tels les réseaux sémantiques et les frames.

En appliquant le principe de résolution et par le mécanisme d'unification, Prolog offre un mécanisme d'inférence puissant, ce qui allège sensiblement la tâche du programmeur.

Enfin, le contrôle de la recherche effectué par Prolog correspond à une recherche en profondeur d'abord (dans l'arbre de recherche ET/OU), balayant les règles de haut en bas et les littéraux de gauche à droite. Là encore, c'est une facilité mais la façon stéréotypée et statique du travail de l'interpréteur se révèle parfois un handicap, particulièrement dans les applications où l'expression d'une heuristique est nécessaire. Dans ce dernier cas, il faut "truquer" le fonctionnement de l'interpréteur, en particulier en jouant sur l'ordre des règles et en utilisant les primitives extra-logiques, ce qui rend souvent le programme difficile à maîtriser, même pour son concepteur.

CONCLUSION

Développer un réel système expert médical est un travail long et complexe qui sortirait du cadre d'un mémoire de fin d'étude; c'est la raison pour laquelle, dès le début, le but a été de réaliser un prototype à caractère expérimental. Cela signifie que, si le système n'est pas destiné à servir en pratique quotidienne, son développement est néanmoins basé sur une étude réaliste de l'existant et doit mener à un produit fini qui puisse faire entrevoir les possibilités d'un système expert.

C'est dans cet ordre d'idée qu'ont été définis les objectifs du projet: susciter l'intérêt du personnel soignant, aborder une nouvelle approche informatique (système expert), systématiser et expliciter des connaissances et une pratique médicales.

Pour y arriver, nous avons choisi un sujet bien délimité où une aide à la décision et au diagnostic peut se révéler utile vu le grand nombre d'alternatives et le caractère incomplet de l'information: la mise au point d'un patient comateux en urgence.

Les deux éléments de départ étaient donc une situation existante (le sujet) et une technique informatique (système expert). Il faut remarquer que ces deux pôles du développement ne sont pas complètement indépendants en ce sens que, dès le début, l'accent a été mis sur l'étude des connaissances et des raisonnements, deux concepts clés dans la mise au point d'un système expert.

Il a donc fallu étudier d'une part la connaissance scientifique, c'est-à-dire la connaissance contenue dans les livres et revues de médecine et d'autre part la connaissance empirique provenant de l'analyse de la pratique quotidienne basée sur l'expérience. Très vite, il nous est apparu que les connaissances et les raisonnements mis en jeu étaient multiples et complexes. La principale difficulté a été à ce stade de les identifier et de les systématiser. Cette étape a demandé un travail très important, nécessitant une étude minutieuse de la littérature et des dossiers médicaux ainsi qu'un grand nombre d'interviews du personnel soignant. Le résultat est l'exposé du sujet qui fait l'objet du cinquième chapitre. Les critères de décision et de diagnostic ont été explicités de façon à préparer le travail pour la représentation des connaissances. Par ailleurs, deux fonctions principales ont été identifiées: modéliser un diagnostic sur base d'information incomplète (fonction déclarative) et

rendre compte de la démarche, des gestes urgents à poser dans la mise au point d'un patient comateux (fonction procédurale).

Parallèlement, l'étude comparative des techniques de représentation des connaissances et de formalisation du raisonnement ont permis de cerner leurs possibilités et lacunes respectives, et cela en regard de l'application qui nous occupe. C'est ainsi que les possibilités offertes par les systèmes experts ont permis d'ajouter quatre nouvelles fonctionnalités au futur prototype à savoir une représentation explicite de la connaissance, la réalisation d'une interface permettant un dialogue proche de celui du langage naturel, l'accès donné à l'utilisateur aux connaissances et la capacité de donner des justificatifs aux questions posées et aux résultats fournis.

Comme on le voit, nous avons préféré prendre une optique assez large dans le but d'offrir à ce travail une assise suffisamment solide pour l'étendre et le raffiner dans le futur. C'est pourquoi, nous parlons de fonction diagnostique sur base de données incomplètes et non de diagnostic de coma. De même, rendre la connaissance explicite, réaliser une interface et un système d'explication sont des fonctions parfaitement transposables parce que générales. Les caractéristiques propres à l'application ont été rassemblées dans la modélisation de la démarche procédurale.

Avant de réaliser ces spécifications, nous avons commencé par par une phase de conceptualisation pour définir les concepts en jeu (diagnostic, antécédent, signe clinique, ...), leurs valeurs de vérité et les relations qui pouvaient les lier. La phase de formalisation a ensuite permis de choisir un mode de représentation des concepts (une variante du classique triplet <objet, attribut, valeur>) et des relations (les règles de production).

Ces deux phases préliminaires ont permis d'établir un cadre de travail pour la réalisation des spécifications. Le système a été conçu en trois phases successives: les gestes urgents, l'évocation d'hypothèses diagnostiques et l'établissement de diagnostics. La première a été modélisée par un algorithme de décision qui propose des mesures à prendre et enregistre les premières données. A partir de celles-ci, on procède à l'évocation d'hypothèses par l'utilisation des règles de diagnostic en chaînage avant. Enfin, les hypothèses retenues sont évaluées par l'emploi

des règles correspondantes en chaînage arrière.

Chaque règle est écrite de manière déclarative et explicite de façon à pouvoir y accéder facilement et ainsi de rendre la base de connaissances ouverte à l'utilisateur. De plus, cela permet d'afficher la règle incriminée lors d'une question ou d'un résultat en guise d'explication et de faciliter la réalisation d'une interface proche du langage naturel.

L'implémentation, réalisée en Turbo Prolog sur un micro-ordinateur de 640K de mémoire centrale, a mis en exergue la facilité offerte par ce logiciel pour la représentation des règles. Par contre, le mode stéréotypé de fonctionnement du moteur d'inférence Prolog s'est révélé un obstacle à la modélisation du raisonnement (chaînage avant et chaînage arrière). De plus, il est apparu que le Turbo Prolog sur micro-ordinateur ne convenait pas parfaitement à la taille du projet. Des difficultés de performances (essentiellement de dépassement de capacité de pile) sont apparues et ont nécessité une révision des programmes dans le sens d'une économie de mémoire (récursion, points de backtracking) parfois au dépend de la lisibilité.

Ce dernier point mériterait une étude plus approfondie des performances. D'autre part, des modules d'acquisition des connaissances, de validation de la base de connaissances et de modélisation du raisonnement évolutif (logique non monotone) pourraient constituer des extensions à ce travail.

Au delà des clichés, la réalisation de ce mémoire nous a permis de découvrir une approche prometteuse dans l'aide à la décision et au diagnostic: les systèmes experts. Par la réalisation d'un prototype expérimental, nous avons découvert, à travers toutes les étapes de développement, une approche différente de l'informatique "classique" (algorithmique). Formaliser les connaissances et le raisonnement et définir un mode adéquat de représentation ont été les principales difficultés.

Il est probable que les systèmes experts médicaux se développeront dans les années à venir et il est à souhaiter qu'en fin de compte, la qualité des soins s'en trouvera améliorée. Par ce prototype, nous espérons simplement stimuler l'intérêt du personnel soignant pour ce type d'outil et ouvrir la voie pour le développement d'un système plus complet qui puisse être testé sur des cas réels. Dans cette perspective, il pourrait

être d'un intérêt didactique pour les médecins et infirmières débutants et, peut-être, une réelle aide à la décision et au diagnostic.

ANNEXE

```

/*****
/*
/*          BASE DE CONNAISSANCES
/*          -----
/*
/*****

/*-----*/
/* Cette base de connaissances reprend les connaissances médicales
/*   du système sous forme de règles de production:
/*
/*          SI liste-et de conditions
/*          ALORS conclusion
/*
/*          où le format de la règle est le suivant:
/*
/*   règle ::= rg(numéro de règle,et-liste de conditions,conclusion)
/*   condition ::= triplet
/*   conclusion ::= triplet
/*   triplet ::= (type d'objet, nom, valeur)
/*
/* type prend une des valeurs suivantes:  do (donnée d'ordre général)
/*                                          at (antécédent)
/*                                          sy (symptôme)
/*                                          sc (signe clinique)
/*                                          ec (résultat d'examen com-
/*                                          plémentaire)
/* (N.B.:un signe est une information d'un des types suivants: do,at,
/*      sy,sc ou ec)
/*                                          sd (syndrome = ensemble de
/*                                          signes)
/*                                          di (diagnostic)
/*                                          tr (traitement)
/*
/* valeur représente une valeur de vérité qui peut être:
/*      pour un signe
/*      ou un syndrome:  vr (vrai), fa (faux),
/*                        de (déductible)
/*      pour un traitement: in (indiqué), ci (contre-
/*                        indiqué)
/*      pour un diagnostic: co (complet, tous les
/*                        signes)
/*                        ev (évoqué),
/*                        su (suspecté),
/*                        pr (prouvé),
/*                        ex (exclu),
/*      pour un examen
/*      complémentaire ou
/*      une donnée numérique: résultat en termes
/*                        qualitatifs (en toutes
/*                        lettres)
/*-----*/

```

PREDICATES

```

    domaine_enum(string,string,stringlist) /*domaine dont les valeurs sont à
                                             énumérer*/
    domaine_std(string,stringlist)         /*domaine standard*/
    rg(integer,liste_triplets,triplet)    /*règle*/
    liste_dia_est(stringlist)             /*liste des diagnostics*/
    type_domaine(string,string,string)    /*type de domaine d'un triplet*/

```

CLAUSES


```
/******  
/*Données déductibles*/  
/******
```

```
/*Signes de latéralisation*/
```

```
/*-----*/
```

```
rg(-1,[tr(sc,hémiplégie,vr)],  
tr(sd,"signes de latéralisation",pr)).
```

```
rg(-2,[tr(sc,"signe de Babinski unilatéral",vr)],  
tr(sd,"signes de latéralisation",pr)).
```

```
rg(-3,[tr(sc,anisocorie,vr)],  
tr(sd,"signes de latéralisation",pr)).
```

```
rg(-4,[tr(sc,"réflexe photomoteur absent d'un côté",vr)],  
tr(sd,"signes de latéralisation",vr)).
```

```
/*Signes d'engagement*/
```

```
/*-----*/
```

```
rg(-5,[tr(sc,"respiration de Cheyne-Stokes",vr),  
tr(sc,anisocorie,vr),  
tr(sc,"réflexe photomoteur absent d'un côté",vr),  
tr(sc,"hypertonie bilatérale",vr)],  
tr(sd,"signes d'engagement",vr)).
```

```
/*Signes d'hypertension intracrânienne*/
```

```
/*-----*/
```

```
rg(-6,[tr(sy,"notion de céphalées",vr),  
tr(sc,"fréquence cardiaque",diminuée),  
tr(sc,"tension artérielle",augmentée),  
tr(sc,"fond d'oeil","stase papillaire")],  
tr(sd,"signes d'hypertension intracrânienne",vr)).
```

```
rg(-7,[tr(sd,"signes d'engagement",de)],  
tr(sd,"signes d'hypertension intracrânienne",vr)).
```

```
/*Stigmates de l'alcoolisme*/
```

```
/*-----*/
```

```
rg(-8,[tr(sc,"angiomes stellaires",vr),  
tr(sc,hépatomégalie,vr),  
tr(ec,"biologie:signes d'alcoolisme(gamma-gt,macrocytose,...)",vr)],  
tr(sd,"stigmates de l'alcoolisme",vr)).
```

```
/******
```

```
/*Critères de diagnostic*/
```

```
/******
```

```
/*Traumatisme crânien simple*/
```

```
/*-----*/
```

```
rg(1,[tr(sy,"notion de traumatisme",vr),  
tr(sc,"hémorragie auriculaire",vr),  
tr(sc,"hémorragie nasale",vr),  
tr(sc,"plaie du cuir chevelu",vr),  
tr(sc,hémotympan,vr),  
tr(ec,"rx crâne",fracture)],  
tr(di,"traumatisme crânien simple",co)).
```



```

rg(2,[tr(sy,"notion de traumatisme",vr)],
    tr(di,"traumatisme crânien simple",ev)).

rg(3,[tr(sc,"hémorragie auriculaire",vr)],
    tr(di,"traumatisme crânien simple",su)).

rg(4,[tr(sc,"hémorragie nasale",vr)],
    tr(di,"traumatisme crânien simple",su)).

rg(5,[tr(sc,"hémotympan",vr)],
    tr(di,"traumatisme crânien simple",su)).

/*Traumatisme crânien avec hématome intracrânien compressif*/
/*-----*/
rg(6,[tr(sy,"notion de traumatisme",vr),
    tr(sy,"intervalle lucide entre l'accident et la perte de connaissance",vr),
    tr(sc,"hémorragie auriculaire",vr),
    tr(sc,"hémorragie nasale",vr),
    tr(sc,"plaie du cuir chevelu",vr),
    tr(sc,"hémotympan",vr),
    tr(ec,"rx crâne",fracture),
    tr(ec,"scanner cérébral","image d'hématome intracrânien")],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",co)).

rg(7,[tr(sy,"notion de traumatisme",vr)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",ev)).

rg(8,[tr(sc,"hémorragie auriculaire",vr)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",ev)).

rg(9,[tr(sc,"hémorragie nasale",vr)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",ev)).

rg(10,[tr(sc,"hémotympan",vr)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",ev)).

rg(11,[tr(sy,"intervalle lucide entre l'accident et la perte de connaissance",vr)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",su)).

rg(12,[tr(sd,"signes de latéralisation",de)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",su)).

rg(13,[tr(sd,"signes d'hypertension intracrânienne",de)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",su)).

rg(14,[tr(ec,"rx crâne",fracture)],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",su)).

rg(15,[tr(ec,"scanner cérébral","image d'hématome intracrânien")],
    tr(di,"traumatisme crânien avec hématome intracrânien compressif",pr)).

/*Coma éthylique*/
/*-----*/
rg(16,[tr(at,"patient alcoolique",vr),
    tr(at,"notion d'excès d'alcool",vr),
    tr(sd,"stigmates de l'alcoolisme",de),
    tr(sc,"température,diminuée"),
    tr(sc,"tension artérielle",diminuée),
    tr(sc,"haleine","odeur d'alcool"),
    tr(ec,"alcoolémie,augmentée")],
    tr(di,"coma éthylique",co)).

rg(17,[tr(at,"patient alcoolique",vr)],
    tr(di,"coma éthylique",ev)).

```



```
rg(18,[tr(sd,"stigmates de l'alcoolisme",de)],  
      tr(di,"coma éthylique",ev)).
```

```
rg(19,[tr(sc,température,diminuée)],  
      tr(di,"coma éthylique",su)).
```

```
rg(20,[tr(sc,"tension artérielle",diminuée)],  
      tr(di,"coma éthylique",su)).
```

```
rg(21,[tr(sy,"notion d'excès d'alcool",vr)],  
      tr(di,"coma éthylique",su)).
```

```
rg(22,[tr(sc,"haleine:odeur d'alcool",vr)],  
      tr(di,"coma éthylique",su)).
```

```
rg(23,[tr(ec,alcoolémie,augmentée)],  
      tr(di,"coma éthylique",su)).
```

```
/*Liste de diagnostics*/
```

```
/*-----*/
```

```
liste_dia_est(["traumatisme crânien simple",  
              "traumatisme crânien avec hématome intracrânien compressif",  
              "coma éthylique"]).
```

```
/*DOMAINES DE VALEUR DES TRIPLETS
```

```
-----*/
```

```
/*Domaines standard*/
```

```
/*numérique (conversion en termes qualitatifs)*/
```

```
domaine_std(num,["normal(e)", "diminué(e)", "augmenté(e)"]).
```

```
/*Définition des domaines standards attachés aux triplets*/
```

```
type_domaine(sc,"fréquence cardiaque",num).
```

```
type_domaine(sc,"tension artérielle",num).
```

```
type_domaine(sc,température,num).
```

```
/*Domaines dont les valeurs sont à énumérer*/
```

```
domaine_enum(ec,"scanner cérébral",[normal,"image d'hématome intracrânien"]).
```

```
domaine_enum(ec,"alcoolémie",[normale,augmentée]).
```

```
domaine_enum(ec,"rx crâne",[normal,fracture]).
```

```
domaine_enum(sc,haleine,[normale,"odeur d'alcool","odeur d'acétone"]).
```

```
domaine_enum(ec,"fond d'oeil",[normal,"rétinopathie diabétique",  
                               "stase papillaire"]).
```

```

/*****
/*
/*          MOTEUR D'INFERENCE (Chainage avant)          */
/*          -----          */
/*          */
/*          */
*****/

```

```

/*-----*/
/*Phase d'évocation d'hypothèses diagnostiques*/

```

PREDICATES

```

ch_avant                                /*chainage avant*/
chercher_di(string,string,string,liste_N,liste_N)/*chercher les règles de
                                                    diagnostic*/
chercher_de(string,string,string,liste_N,liste_N)/*chercher les règles des
                                                    données à déduire*/

```

CLAUSES

```

ch_avant :-
    chercher_di(di,_,_,[0],Liste_di),
    chercher_de(_,_,vr,Liste_di,Liste_tot),
    traiter_rg(ch_avt,Liste_tot,_).

```

```

/*Recherche de règles de diagnostic*/
chercher_di(di,_,_,Liste1,Liste2):-
    rg(N,_,tr(di,_,_)),
    not(membre(N,Liste1)),
    chercher_di(di,_,_,[N>Liste1],Liste2).

```

```

/*il n'y a plus de règle valide, l'affectation finale peut se faire*/
chercher_di(_,_,_,Liste1,Liste2):-
    Liste1 = Liste2.

```

```

/*Recherche de règles de donnée déductible*/
chercher_de(_,_,vr,Liste1,Liste2):-
    rg(N,_,tr(_,_,vr)),
    not(membre(N,Liste1)),
    chercher_de(_,_,vr,[N>Liste1],Liste2).

```

```

/*il n'y a plus de règle valide, l'affectation finale peut se faire*/
chercher_de(_,_,_,Liste1,Liste2):-
    Liste1 = Liste2.

```



```

/*****
/*
/*          MOTEUR D'INFERENCE (chainage arrière)          */
/*          -----          */
/*          */
/*****

```

```

/*-----*/

```

PREDICATES

```

ch_arriere          /*initialiser la liste de diagnostics et
                    lancer l'évaluation par chainage arrière*/
evaluer(string)     /*évaluer un diagnostic*/

eval_etat(string,string,res) /*évaluer l'état d'un diagnostic*/

rech_diag(stringlist) /*recherche de diagnostics à évaluer en les
                      extrayant un à un de la liste*/

```

CLAUSES

```

/*Recherche de diagnostics*/
/*-----*/

```

```

ch_arriere:-
    liste_dia_est(Liste),
    rech_diag(Liste).

```

```

rech_diag([]).
rech_diag([Dia:Reste]):-
    evaluer(Dia),
    rech_diag(Reste).

```

```

/*Evaluation de diagnostics*/
/*-----*/

```

```

/*si le diagnostic a pour état co (tableau complet)
                                pr (prouvé)
                                ex (exclu)
    alors il n'est plus nécessaire de l'évaluer.*/

```

```

evaluer(Dia):-
    ft(di,Dia,co),!.
evaluer(Dia):-
    ft(di,Dia,pr),!.
evaluer(Dia):-
    ft(di,Dia,ex),!.

```

```

/*sinon, si il a pour état su (suspecté), évaluer tous les signes, marquer
le diagnostic comme évalué et vérifier si on peut le faire passer dans
l'état co (complet), pr (prouvé) ou ex(exclu)*/

```

```

evaluer(Dia):-
    ft(di,Dia,su),!,
    eval_etat(Dia,co,Res),
    maj_data(di,Dia,co,Res),
    eval_etat(Dia,pr,Res2),

```

```
maj_data(di,Dia,pr,Res2),
eval_etat(Dia,ex,Res3),
maj_data(di,Dia,ex,Res3).
```

```
/*sinon, si il a pour état ev (évoqué), évaluer tous les signes, marquer
le diagnostic comme évalué et vérifier si on peut le faire passer dans
l'état co (complet), su (suspecté), pr (prouvé) ou ex (exclu)*/
```

```
evaluer(Dia):-
```

```
ft(di,Dia,ev),!,
eval_etat(Dia,co,Res),
maj_data(di,Dia,co,Res),
eval_etat(Dia,su,Res2),
maj_data(di,Dia,su,Res2),
eval_etat(Dia,pr,Res3),
maj_data(di,Dia,pr,Res3),
eval_etat(Dia,ex,Res4),
maj_data(di,Dia,ex,Res4).
```

```
/*sinon, si il n'a pas encore d'état attribué, essayer d'abord d'exclure
ce diagnostic et si ce n'est pas possible, évaluer tous les signes et
vérifier si on peut le faire passer dans l'état co (complet),
ev (évoqué), su (suspecté), ou pr (prouvé)*/
```

```
evaluer(Dia):-
```

```
not(ft(di,Dia,_)),!,
eval_etat(Dia,ex,Res),
maj_data(di,Dia,ex,Res),
eval_etat(Dia,co,Res2),
maj_data(di,Dia,co,Res2),
eval_etat(Dia,ev,Res3),
maj_data(di,Dia,ev,Res3),
eval_etat(Dia,su,Res4),
maj_data(di,Dia,su,Res4),
eval_etat(Dia,pr,Res5),
maj_data(di,Dia,pr,Res5).
```

```
/*Evaluer l'état d'un diagnostic*/
```

```
/*-----*/
```

```
/*si le diagnostic est déjà passé dans un état définitif (co,pr ou ex),
alors l'évaluation de tout autre état devient superflue et est considérée
comme un échec*/
```

```
eval_etat(Dia,_,Res):-
```

```
ft(di,Dia,co),Res = échec,!.
eval_etat(Dia,_,Res):-
```

```
ft(di,Dia,pr),Res = échec,!.
eval_etat(Dia,_,Res):-
```

```
ft(di,Dia,ex),Res = échec,!.

/*sinon, évaluation de l'état*/
```

```
eval_etat(Dia,Etat,Res):-
```

```
chercher_rg(di,Dia,Etat,[0],Liste),
traiter_rg(ch_arr,Liste,Res).
```



```

/*****
/*
/*                               INTERFACE                               */
/*                               -----                               */
/*
*****/

```

```

/*-----*/
/*Ce module gère les opérations d'interface avec l'utilisateur: dialogue
   en un sous ensemble du langage naturel, menus,...*/

```

```

INCLUDE "tdoms.pro"
INCLUDE "tpreds.pro"
INCLUDE "menu.pro"

```

PREDICATES

```

    domaine(triplet,string,stringlist) /*domaine d'un triplet*/
    ecrire_type(string)                 /*écrire type en langage naturel*/
    ecrire_valeur(string)               /*écrire valeur en langage naturel*/
    encore                             /*artifice de boucle*/
    exec_question(string,stringlist,triplet,res) /*poser une question concer-
                                                    nant un triplet*/

```

CLAUSES

/*UTILITAIRES

```

-----*/

```

```

/*artifice pour éliminer les valeurs illégales dans un menu*/

```

```

    encore.
    encore:-
        encore.

```

```

/*interface pour écrire une question ou un résultat*/

```

```

    ecrire_valeur(vr).
    ecrire_valeur(in):-
        write(" indiqué").
    ecrire_valeur(ci):-
        write(" contre-indiqué").
    ecrire_valeur(co):-
        write(" tableau complet").
    ecrire_valeur(ev):-
        write(" évoqué").
    ecrire_valeur(su):-
        write(" suspecté").
    ecrire_valeur(pr):-
        write(" prouvé").
    ecrire_valeur(ex):-
        write(" exclu").
    ecrire_valeur(Texte):-
        write(Texte).

    ecrire_type(do).
    ecrire_type(ec).
    ecrire_type(at):-
        write(" dans les antécédents").
    ecrire_type(sy):-
        write(" dans la symptomatologie").
    ecrire_type(sc):-
        write(" à l'examen clinique").

```

```

/*SELECTION D'UN TYPE DE QUESTION*/
question(tr(Type,Nom,Valeur),Res):-
    domaine(tr(Type,Nom,Valeur),Type_q,Liste_valeurs),!,
    exec_question(Type_q,Liste_valeurs,tr(Type,Nom,Valeur),Res).

/*PROCEDURES DE RECHERCHE DE DOMAINES*/
/*Domaines standard definis*/
domaine(tr(Type,Nom,Valeur),Type_q,Liste_valeurs):-
    type_domaine(Type,Nom,Type_domaine),!,
    domaine_std(Type_domaine,Liste_valeurs),
    Type_q = menu.

/*Domaines dont les valeurs sont à énumérer*/
domaine(tr(Type,Nom,Valeur),Type_q,Liste_valeurs):-
    domaine_enum(Type,Nom,Liste_valeurs),!,
    Type_q = menu.

/*Par défaut, les valeurs possibles pour un triplet sont vrai (présent)
et faux (absent), c'est-à-dire une question de type simple*/
domaine(tr(Type,Nom,_),Type_q,_):-
    not(type_domaine(Type,Nom,_)),
    not(domaine_enum(Type,Nom,_)),
    Type_q = simple.

/*MENUS-QUESTIONS
-----*/

/*menu le plus simple: exec_question du type oui/non*/
/*N.B.:une question simple est toujours posée de manière positive*/
exec_question(simple,_,tr(Type,Nom,Valeur),Res):-
    encore,
    makewindow(1,7,135,"QUESTION:",1,40,8,40),
    write(Nom),nl,
    ecrire_type(Type),
    write("(o/n)"),nl,
    readchar(Rep),
    removewindow,
    anal_rep(Rep,Type,Nom,Valeur,Res).

/*menu*/
exec_question(menu,Liste_valeurs,tr(Type,Nom,Valeur),Res):-
    makewindow(2,7,7,"MENU",1,40,3,40),
    write("RET pour choisir, flèche pour changer"),
    menu(4,40,7,135,Liste_valeurs,Nom,1,Choix),
    removewindow,
    extraire(Choix,Liste_valeurs,Valeur_extraite),
    anal_val(Valeur,Valeur_extraite,Res),
    anal_rep('o',Type,Nom,Valeur_extraite,Res).

/*MENUS DE DIALOGUE
-----*/

```



```
menu_prin:-
    encore,
    makewindow(1,7,7,"MENU PRINCIPAL",0,0,25,80),
    /*positionner le curseur*/
    nl,nl,nl,nl,nl,
    write("      1°Interrogation de la base de connaissances"),nl,nl,
    write("      2°Démarche en urgence"),nl,nl,
    write("      3°Module diagnostique"),nl,nl,
    write("      4°Impression du rapport"),nl,nl,
    write("      5°Sortie"),nl,nl,nl,nl,
    write("      Votre choix?"),
    readchar(Rep),
    anal_rep_menu_pr(Rep).

anal_rep_menu_pr('1'):-
    menu_database.

anal_rep_menu_pr('3'):-
    ch_arriere.

anal_rep_menu_pr('5'):-
    exit.
```

```

/*****
/*
/*          UTILITAIRES          */
/*          -----          */
/*
/*****

```

PREDICATES

```

chercher_rg(string,string,string,liste_N,liste_N)
/*chercher les numéros des règles répondant
au triplet donné et les stocker dans une
liste*/

effacerft(string,string,string) /*retirer, s'il en existe un, le
fait décrit par le triplet donné*/

eval_crit(string,integer,liste_triplets,string,res)
/* évaluer, en fonction d'un paramètre
(chainage arrière ou chainage avant) les
critères (d'un diagnostic dans un certain
etat) contenus dans la liste extraite de la
règle dont le numéro est num et mettre le
résultat de l'évaluation dans res */

maj_data(string,string,string,res) /*mise à jour d'une donnée déductible
ou de l'état d'un diagnostic en
fonction du résultat fourni*/

membre(integer,liste_N) /*déterminer si un entier appartient à une
liste d'entiers*/

superieur(string,string) /*un etat est supérieur à un autre s'il
contient plus d'information*/

test_eval(string,string,string,integer,liste_triplets,liste_N)
/*évaluation (ou non selon le résultat du
test) d'une règle en chainage avant*/

testfin(string,res,liste_N,res)/*test de fin de traitement d'une liste de
règles*/

traiter_rg(string,liste_N,res)/*traiter les règles retenues dans la liste
selon le paramètre de traitement fourni et
donner le résultat du traitement*/

verif_eval(string,string,string,res)/*vérifier si la règle dont la conclusion
est tr(type,nom,valeur) vaut la peine
d'être évaluée*/

```

CLAUSES

```

/*Membre d'une liste
-----*/
membre(Elt,[Elt:_]).
membre(Elt,[_:Reste]):-
membre(Elt,Reste).

/*Extraire le n'ième élément d'une liste
-----*/
extraire(0,_,nul).
extraire(1,[Premier:_],Premier).
extraire(N,[_:Reste],Valeur):-
K = N - 1,
extraire(K,Reste,Valeur).

```

```

/*Chercher les numéros des règles répondant au triplet
-----*/
chercher_rg(Type,Nom,Valeur,Liste1,Liste2):-

```



```

rg(N,_,tr(Type,Nom,Valeur)),
not(membre(N,Liste1)),
chercher_rg(Type,Nom,Valeur,[N>Liste1],Liste2).

```

```

/*il n'y a plus de règle valide, l'affectation finale peut se faire*/
chercher_rg(____,Liste1,Liste2):-
    Liste1 = Liste2.

```

```

/*Traiter les règles retenues
-----*/

```

```

traiter_rg(____,[0],échec)./*le 0 est le signe de fin de liste*/

```

```

/*chainage arrière*/

```

```

traiter_rg(ch_arr,[N:Reste],Res_trt):-
    rg(N,Liste,tr(____,Etat)),
    eval_crit(ch_arr,N,Liste,Etat,Res),
    testfin(ch_arr,Res,Reste,Res_trt).

```

```

/*chainage avant*/

```

```

traiter_rg(ch_avt,[N:Reste],_):-
    rg(N,Liste,tr(Type,Nom,Valeur)),
    test_eval(Type,Nom,Valeur,N,Liste,Reste).

```

```

/*chainage arrière*/

```

```

testfin(ch_arr,Res,____):-
    free(Res)./*on s'arrête à la première règle dont l'évaluation est
    positive*/
testfin(ch_arr,échec,Reste,Res_trt):-
    traiter_rg(ch_arr,Reste,Res_trt)./*sinon, on évalue une autre règle applicable*/

```

```

/*test d'évaluation en chaînage avant d'une règle de donnée déductible
ou d'une règle de diagnostic*/

```

```

test_eval(Type,Nom,Valeur,N,Liste,Reste):-
    verif_eval(Type,Nom,Valeur,Res_ver),
    free(Res_ver),/*test positif, on peut évaluer la règle*/
    eval_crit(ch_avt,N,Liste,Valeur,Res),
    maj_data(Type,Nom,Valeur,Res),
    traiter_rg(ch_avt,Reste,____).

```

```

test_eval(Type,Nom,Valeur,____,Reste):-
    verif_eval(Type,Nom,Valeur,échec),/*test négatif, passer à la règle
    suivante*/
    traiter_rg(ch_avt,Reste,____).

```

```

/*vérification d'utilité d'application d'une règle*/

```

```

/*dans le cas d'une donnée déductible*/

```

```

verif_eval(Type,Nom,vr,____):-
    not(ft(Type,Nom,vr)),!./*règle applicable si cette donnée n'est pas déjà
    enregistrée dans la base de faits*/
verif_eval(____,vr,échec)./*règle non applicable sinon*/

```

```

/*dans le cas d'une règle de diagnostic*/

```

```

verif_eval(di,Dia,____):-
    not(ft(di,Dia,____)),!./*règle applicable si cette donnée n'est pas déjà
    enregistrée dans la base de faits*/

```

```

verif_eval(di,Dia,Etat,____):-

```

```

    ft(di,Dia,Etatft),
    Etatft <> "ex",
    Etatft <> "pr",
    Etatft <> "co",
    superieur(Etat,Etatft),!.
verif_eval(di,_,_,échec).

```

```

superieur(co,ev).
superieur(ex,ev).
superieur(su,ev).
superieur(pr,ev).
superieur(ex,su).
superieur(pr,su).
superieur(co,su).

```

```

/*Evaluation d'une liste de critères
-----*/

```

```

/*1)Modules propres au chaînage avant*/

```

```

/*l'évaluation s'arrête dès qu'un élément s'avère négatif*/
eval_crit(ch_avt,_,_,Valeur,Res):-
    not(free(Res)),/*(Res=échec)*/
    eval_crit(ch_avt,_,[],Valeur,_)./*arrêt de l'évaluation*/

```

```

/*2)Modules communs aux deux modes de recherche (chaînage arrière et
chaînage avant)*/

```

```

eval_crit(_,_,[],_,_).

```

```

/*voir si le critère à évaluer n'est pas déjà enregistré dans la base de
faits*/
eval_crit(C,N,[tr(Type,Nom,Valeur):Reste],Etat,Res):-
    ft(Type,Nom,Valeurft),
    anal_val(Valeur,Valeurft,Res),/*voir si la valeur de vérité du fait déjà
                                enregistré est la même que celle du fait
                                recherché*/
    eval_crit(C,N,Reste,Etat,Res).

```

```

/*3)Modules propres au chaînage arrière*/

```

```

/*si on évalue tout le tableau d'un diagnostic (état co), il faut
envisager tous les éléments de la liste
sinon, l'évaluation s'arrête dès qu'un élément s'avère négatif*/
eval_crit(ch_arr,_,_,Etat,Res):-
    not(free(Res)),/*(Res=échec)*/
    Etat <> "co",
    eval_crit(ch_arr,_,[],Etat,_)./*arrêt de l'évaluation*/

```

```

/*pour évaluer un critère (fait élémentaire),
voir si le critère à évaluer n'est pas déjà enregistré dans la base de
faits*//*(cfr modules communs)*/

```

```

/*sinon, si c'est un fait déductible alors le déduire*/
eval_crit(ch_arr,N,[tr(Type,Nom,de):Reste],Etat,Res):-
    chercher_rg(Type,Nom,vr,[0],Liste),
    traiter_rg(ch_arr,Liste,Res),
    maj_data(Type,Nom,vr,Res),

```



```

eval_crit(ch_arr,N,Reste,Etat,Res).

/*sinon, questionner l'utilisateur*/
eval_crit(ch_arr,N,[tr(Type,Nom,Valeur):Reste],Etat,Res):-
    question(tr(Type,Nom,Valeur),Res),
    eval_crit(ch_arr,N,Reste,Etat,Res).

/*Gestion de la base de faits
-----*/

/*analyse du résultat de l'évaluation d'un triplet
(question posée à l'utilisateur)*/
/*N.B.: 1°la question (o/n) concernant un fait est toujours posée de manière
positive même si le fait à évaluer a la valeur de vérité fa.
2°la réponse à une question menu est considérée comme positive pour
l'option choisie et n'est donc jamais négative*/

/*pas de réponse de l'utilisateur*/
anal_rep('o',_,_,nul,échec).
/*réponse positive pour ce triplet*/
anal_rep('o',Type,Nom,Valeur,_):-
    !,assertz(ft(Type,Nom,Valeur)).
/*réponse négative (à une question du type oui/non)*/
anal_rep('n',Type,Nom,Valeur,Res):-
    !,assertz(ft(Type,Nom,fa)),
    anal_val(Valeur,fa,Res).

/*comparaison de deux valeurs*/
anal_val(Valeur,Valeur,_):-!.
anal_val(Valeur1,Valeur2,Res):-
    Valeur1 <> Valeur2,
    Res = échec.

/*mise à jour d'une donnée déductible ou d'un diagnostic évalué par
chaînage arrière ou par chaînage avant*/
maj_data(Type,Nom,Valeur,Res):-
    free(Res),!,
    effacerft(Type,Nom,_),
    assertz(ft(Type,Nom,Valeur)).

/*sinon, pas de mise à jour*/
maj_data(_,_,_,échec):-!.

effacerft(Type,Nom,_):-
    ft(Type,Nom,_),!,
    retract(ft(Type,Nom,_)).
effacerft(_,_,_).

```

REFERENCES

REFERENCES

1) REFERENCES GENERALES

1. ADLASSNIG K.-P., KOLARZ G. et al.
CARDIAG 1: a computer-assisted diagnostic system on the basis of symbolic logic and its application in internal medicine.
Medical Informatics Europe, 1982.
2. BOUCKAERT A.
Medical diagnosis: are expert systems needed?
Int. J. Bio-Medical Computing, 20, 123-133, 1987.
3. BUCHANAN B.G., DUDA R.O.
Principles of rule based expert systems
HPP Report n° HPP-82-14. August 1982.
4. CHOURAQUI E., FARRENY H.
Modélisation du raisonnement et de la connaissance
T.S.I., vol 4, n°4, 1985.
5. DE DOMBAL F.T., LEAPER D.J.
Computer-aided diagnosis of abdominal pain.
British Medical Journal 2, pp 9-13, 1972.
6. DEKEYSER L.J., KREKELS B. et al.
A medical expert system shell based on logic
Proceedings of the 7th IJCAI, pp. 876-881, 1981
7. DE VRIES P.H., DE VRIES ROBBE P.F.
An overview of medical expert systems
Meth. Inform. Med. Vol 24, n°2, 1985.
8. FAGOT LARGEAULT A.
La simulation du raisonnement médical
La Recherche n°170, pp. 1176-1187, octobre 1985.
9. FIESCHI D., FIESCHI M., BOTTI G., JOUBERT M.
An expert system for tutorial reasoning
Proceedings of the 7th IJCAI, pp. 893-900, 1981
10. FIESCHI M.
Intelligence artificielle en médecine
Masson, 1986.
11. FIESCHI M., JOUBERT M.
Some reflections on the evaluation of expert systems in medicine
Meth. Inform. Med. Vol. 25, n°1, 1986.
12. GALLAIRE H.
La représentation des connaissances
La Recherche n°170, pp. 1240-1248, octobre 1985.

13. GANASCIA J.G.
La conception des systèmes experts
La Recherche n° 170, pp.1142-1151, octobre 1985.
14. GASCUEL O.
Un système expert pour la réalisation de diagnostics
T.S.I., vol.4, n°4, 1985.
15. GLASZIOU P., VERMEIR D.
Information analysis for medical expert systems
Meth. Inform. Med. Vol 23, n°3, 1984.
16. HAGAMEN W. D., GARDY M. et al.
MEDCAT: An interactive computer program for medical
diagnosis, consultation and teaching
National computer conference, 1985.
17. KASSIRER J.P., GORRY G.A.
Clinical problem solving: a behavioral analysis
Annals of Internal Medicine, 89, pp.245-255, 1978.
18. LAURIERE J.L.
Intelligence artificielle
résolution de problèmes par l'Homme et la machine
Eyrolles, 1987.
19. MAGREZ P.
Modèles de raisonnement approché dans le cadre des
systèmes experts médicaux.
Thèse déposée en vue de l'obtention du titre d'agréé de
l'enseignement supérieur, 1985.
20. MAGREZ P.
Intelligence artificielle en médecine
Rev. Med. Brux. 7, 153-161, 1986.
21. MILLER R.A., POPE H.E., MYERS J.D.
Internist I, an experimental computer-based diagnostic
consultant for general internal medicine.
New England Journal of Medicine 307: 468-476 (August 19), 1982
22. MINSKY M.
A framework for representing knowledge
The psychology of computer vision
WINSTON Eds. Mc GRAW-HILL pp 211-277, 1975.
23. PATIL R.S.
Causal representation of patient illness for electrolyte and
acid-bases diagnosis.
Ph.D.;, 1981, Laboratory of computer science
Massachusetts Institute of Technology.
24. PAUKER S.G. et al.
Towards the simulation of clinical cognition: taking a
present illness by computer
The american journal of medicine, 60, pp.981-996, 1976.

25. POPLER H.E.
Heuristic methods for imposing structure on ill structured problems: the structuring of medical diagnostics.
in Artificial Intelligence in Medicine.
Eds. P. SZOLOVITS AAAI Selected Symposium 51.
26. ROBINSON J.A.
A machine-oriented logic based on the resolution principle
J. ACM 12, 23-41, 1965.
27. ROGER F.H.
Médecine et informatique
Maloine, 1979.
28. SHAFER G.
A mathematical theory of evidence
Princeton University Press, 297p, 1976.
29. SHORTLIFFE E.
Computer based medical consultations: MYCIN
American Elsevier, New-York, 1976.
30. SHORTLIFFE E.
Consultations systems for physicians
Readings in artificial intelligence, pp. 323-333
Addison-Wesley, 1984.
31. SUBRAHMANYAM P.A.
The "software engineering" of expert systems: is Prolog appropriate?
IEEE Transactions on software engineering,
vol. SE-11, n°11, november 1985.
32. TEACH R., SHORTLIFFE E.
An analysis of physicians' attitudes
Rule-based expert systems
Addison-Wesley, 1984.
33. VAN BEMMEL J. H.
Formalization of medical knowledge
Meth. Inform. Med. Vol. 25, n°4, 1986.
34. VAN LAMSWEERDE A.
Cours d'intelligence artificielle
FNDP Namur, année académique 1986/1987.
35. WEISS S., KULIKOWSKI C.A.
A model-based method for computer-aided medical decision-making.
Artificial Intelligence 11, 145-172, 1978.
36. WILLIAMS B.T.
Computer aids to clinical decisions
Vol 1, chap. 2. CRC Press, 1982.

37.ZADEH L.A.

Probability measures of fuzzy events
Journal of Mathematical Analysis and
Applications,23,412-427,1968.

II)REFERENCES PROPRES A L'APPLICATION

38.ASKENASI R., EVEN-ADIN D.

Manuel de médecine d'urgence de l'adulte
Editions de l'Université de Bruxelles,
chap.11,pp.170,1982.

39.CAMBIER J., MASSON M., DEHEN H.

Abrégé de Neurologie
Masson,chap.5,pp 112-121,1978.

40.HAROLD FRIEDMAN H.

Manuel de diagnostic médical
Chap.10,pp 353-359
Médecines et Sciences Internationales,1980.

41.ISSELBACHER K.,ADAMS R.

Principles of Internal Medicine
Chap.20,pp 114-121
International Student Edition,1980.

42.NOUILHAT F.,SIMON N.

Comment chercher la cause d'un coma
Revue du Praticien,33,pp 1709-1720,1983.